

## TEKST~SPECIAL

- **Publishing Partner**
- **Eigen fonts ontwerpen**
- **1st Aid hulpprogramma voor 1st Word (+ spellingschecker als Special Disk)**

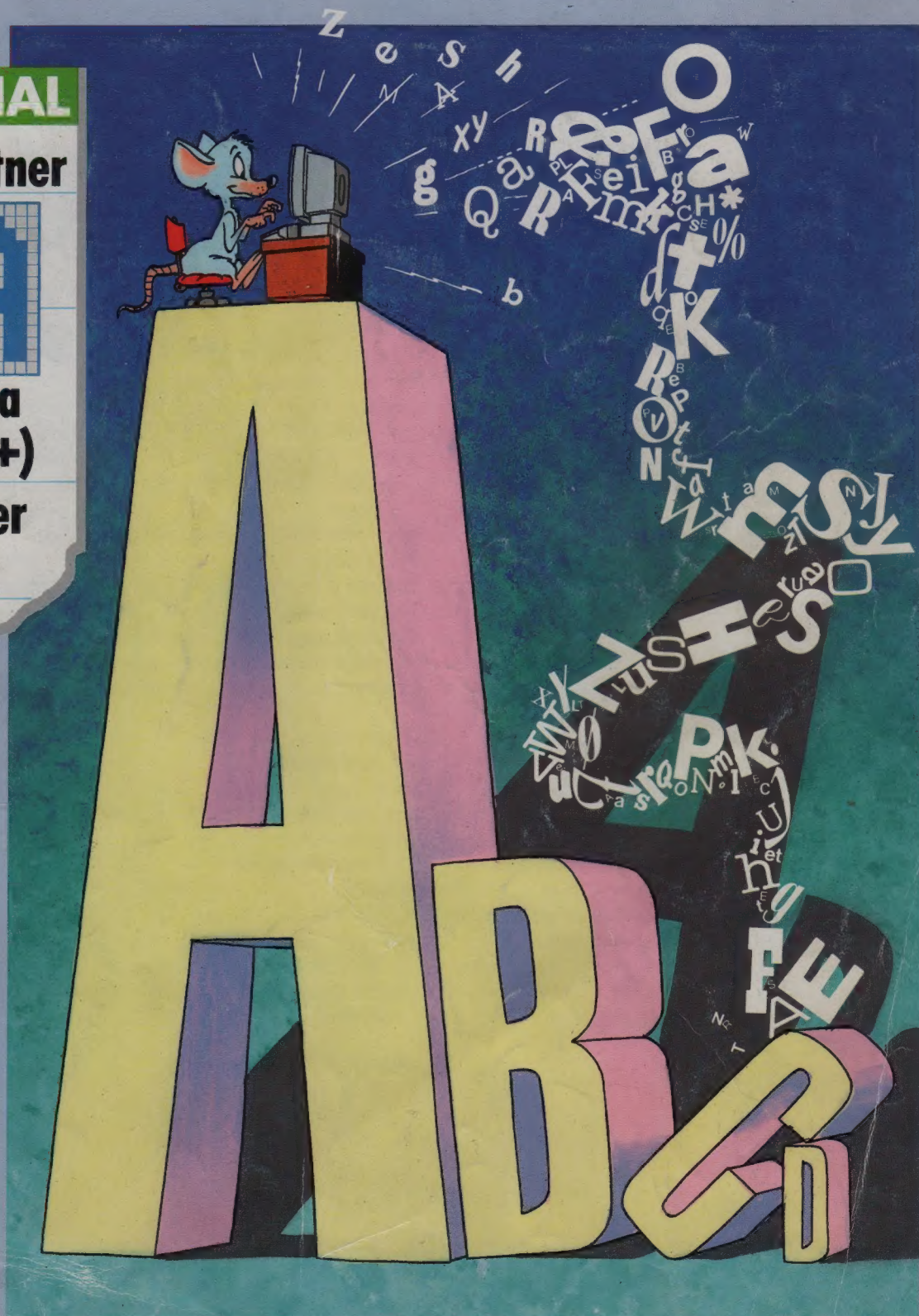


# De marketing plannen van ATARI

### Listing:

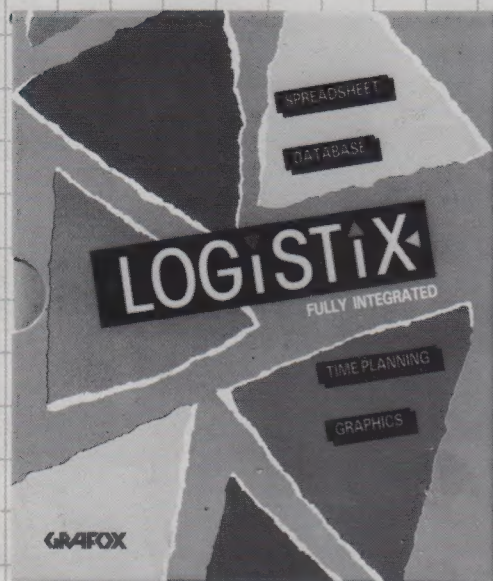
# ASTRO CHALLENGE

# Bouw een accessoires bibliotheek





# Nieuw: Management-Tools voor de ST Logistix.



Logistix is een geïntegreerd software-pakket voor de Atari ST. Met Logistix voegt u aan al u applicaties de dimensie TIJD toe.

Met Logistix heeft u de software-tools in handen welke u in staat stellen om uitgebreide NETWERKPLANNINGEN, VAKANTIEROOSTERS, LESROOSTERS, PRODUCTIE-PLANNINGEN, BEZETTINGSPLANNINGEN VOOR HOTELS, PROJECTPLANNINGEN, TERMIJNPLANNINGEN, KOSTENANALYSES, MEDIAPLANNINGEN, HET MAKEN VAN FLIP-CHARTS etc. te maken.

Logistix bestaat uit de onderdelen: database, tijdplanning, netwerkplanning, spreadsheet en grafieken.

Een schildersbedrijf kan bijvoorbeeld in de database alle relevante gegevens opslaan over de medewerkers, machines, aangenomen werk etc. Deze gegevens dienen als basis voor uw kostprijsberekeningen, voor de capaciteitsplanning, voor de indeling van het werk en u kunt alles dankzij de krachtige grafieken ook nog visueel maken.

Met Logistix kan nu elke manager antwoord geven op vragen zoals:

- \* Hoe lang gaat dit project duren?
- \* Hoeveel kamers zijn er in welke periode nog vrij?
- \* Wat is de bezettingsgraad van mijn vrachtwagens?
- \* Hoeveel personeel en machines zijn er nodig? Wanneer?
- \* Welke produktielijn kan ik voor deze order het beste nemen?
- \* Wat gebeurt er als ik het project uit ga besteden?
- \* Wat zijn de gevolgen nu de heer Brokkemaker ziek is?
- \* Wanneer hebben wij uitzendkrachten nodig?
- \* Moeten we dollars gaan kopen of verkopen?
- \* Hoeveel uitval had lijn 1 deze week?
- \* Hoe ziet mijn liquiditeitsplaatje er nu uit nu taxi 3 niet meer inzetbaar is?

Met Logistix haalt u een management-tool van de jaren negentig in huis. En wat de prijs betreft, die is om te lachen f. 495,-, inclusief b.t.w.

**SOFTPACK**  
INTERNATIONAL

Postbus 6250,  
2702 AG Zoetermeer.  
Tel.: 079-412563





6

show gemist heeft, kan op deze pagina's de schade van de ST heeft niet over de aandacht van soft- en hardware producenten te klagen.



## GfA VRAAGBAAK

9

Heino Kaal

Onze nieuwe rubriek met tips en trucs rond de populairste programmeertaal voor de ST. Initiatiefnemer Heino Kaal gaat van start met een programmeurs- en gebruikersvriendelijke invoerroutine.

## 1ST-AID

13

Wim de Winter

Ook 1ST Wordplus heeft nog wat tekortkomingen. U kunt er bijvoorbeeld niet mee in kolommen afdrukken of een trefwoordenregister samenstellen. Dit hulpprogramma helpt u uit de nood.

## ST SNEL TERUGVERDIEND

17

Alfred Konijnenbelt

START interviewde Wilfried de Graaf, de sales en marketing manager van Atari Benelux. Waarom in vredesnaam een Atari PC?

## PROGRAMMA'S MET KARAKTER

20

Egon Lubbers & Karl Moens

Als u wat uitgekeken raakt op het standaard lettertype van de ST, vindt u hier alle benodigde informatie om voortaan uw eigen karaktersets te gebruiken. Uiteraard krijgt u alle benodigde software (zes listings) kado. Unieke documentatie om altijd bij de hand te hebben.



## PUBLISHING PARTNER LEIDT TOT ECHTSCHIEDING

26

Henri Kruithof

Het eerste desktop publishing programma voor de ST is uit. Henri Kruithof raakte lyrisch over de mogelijkheden op het beeldscherm. De afdrukkwaliteit lijkt echter stiefkindje geworden te zijn.

YVG

See is what you get. It's on the screen just the way it will look when it is more guess work. Just fonts, character sets, even character sets and anywhere. You'll watch an image transform into a looking "piece" right before your eyes as you mix graphics.

up to 144 points

Special Attributes

Backslant  
Bold  
Double Underline  
Italicize  
Light  
Mirror 1011M  
Outline  
Reverse Type  
Shadow  
Strike through  
Tall

Underline  
Upside Down  
Wide

You can mix and match any attribute you desire!

For example:

Tall and Shadow  
Italicize, Outline, and

Import Graphics



You can reduce, enlarge, or "cut" any Degas™ or Neochrome™ picture

To place an order:  
Just call (314) 894-8608 or mail payment to:

SoftLogik Corp  
4129 Old Baumgartner

## SPELLBOUND

### UNIVERSELE SPELLINGSCHHECKER

29

Rob Koopman

Een compleet verslag van de ontwikkeling van onze eigen spellingschecker SPELLBOUND. Dit programma werkt met alle ons bekende programma's samen. Het hart van de software krijgt u als pseudo Modula listing kado. Wie geen zin heeft om zelf te programmeren kan echter ook in onze serie special disks terecht.

## ACCLOAD

32

Wim de Winter

Heeft u ook altijd net de verkeerde accessoires in uw desktop geladen? Dit handige utility maakt het u mogelijk een complete bibliotheek van accessoires aan te leggen en daar op het laatste moment zes uit te kiezen.

## ALLES OVER FLOPPIES

34

Frank Abbing

Hoeveel informatie past er nu precies op een floppy? Heeft uw diskdrive wel de juiste draaisnelheid?

Frank Abbing geeft antwoord op alle vragen en voegt een listing toe om zelf te experimenteren.

## ANIMATOR:

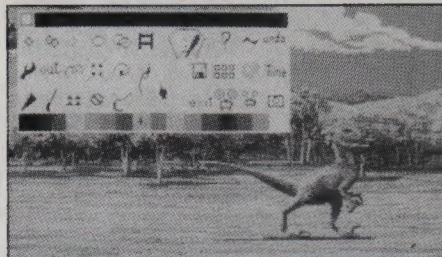
### DE KUNST VAN HET VERVORMEN

38

Robert van der Kroft

De eerste vijf voorplaten van START werden gemaakt door 'Sjors en Sjimmie'-tekenaar Robert van der Kroft.

Het tekenfilm pakket Animator was er de oorzaak van dat hij zijn wekelijkse gag bijna vergat.



## ASTRO CHALLENGE

39

Iban van der Wal

Weer een bewijs voor de snelheid van de 68000. Zelfs een spel in BASIC kan te snel zijn voor uw reflexen!

## MACHINETAAL OP DE ST LINE A ROUTINES

42

Jan Vogel

Ditmaal een uitgebreid voorbeeldprogramma dat u leert hoe u vanuit machinetaal op het scherm kunt tekenen.

## DE MUNITIE VAN DE ST

44

Frank Abbing

Wat is toch de betekenis van de bommen die de ST af en toe op het scherm toont? Dit artikel vertelt u er alles over. Bovendien vindt u op deze pagina's een programma dat het grafisch geweld voortaan omzet in duidelijke tekstinformatie.

## GfA CROSS REFERENCE

48

Peter Witvoet

Een lijvig GfA programma gebruikt veel variabele- en procedur namen. Dit hulpprogramma maakt een overzichtslijst van alle namen die in gebruik zijn, zodat dubbelures voorkomen kunnen worden.

## LEZERSERVICE

50



# DE MARKT

*De meeste produkten die in deze aflevering van "de markt" aan de orde komen, werden geïntroduceerd tijdens de PCM-show. Omdat er vaak prototypes getoond werden, zult u het ditmaal in deze rubriek met wat minder illustraties moeten doen.*

*Overigens viel de publieke belangstelling tijdens de PCM-show tegen. Nederland is wat beursmoe. Maar ook het aantal premières op de beurs was onder de maat: veel bezoekers hielden het na een paar uur voor gezien.*

*Voor ST-liefhebbers waren er twee stands van belang: die van Atari en die van PDS. In tegenstelling tot de rest van de beurs werden hier veel nieuwe produkten gelanceerd. De interessantste stands van de PCM-show dus.*

## MEGA ST VOORLOPIG ZONDER BLITTER

Atari presenteerde haar nieuwe modellen (Mega's en laserprinter) aan het Nederlandse publiek. Er was echter ook een slecht nieuwtje: als de Mega ST straks bij de dealers gedemonstreerd wordt, is die computer niet helemaal dezelfde als die er te koop is. De eerste maanden zullen namelijk alleen Mega ST's zonder blitterchip verkocht worden. Dat is het gevolg van productieproblemen bij General Electric, de chipfabriek die de blitters maakt. Een groot deel van de eerste serie blitterchips werd onlangs door Atari afgekeurd.

Tot medio augustus, wanneer de tweede serie in productie wordt genomen, zijn wereldwijd slechts 2.500 goedgekeurde blitters beschikbaar. Deze zullen volgens sales- en marketingmanager Wilfried de Graaf van Atari worden geplaatst in een beperkt aantal Mega ST's die beschikbaar worden gesteld aan software-ontwikkelaars, dealers en pers.



De blitters zullen zo spoedig mogelijk aan de kopers worden nageleverd, belooft De Graaf. Volgens hem is besloten om de verkoop van Mega ST's niet op te schorten in afwachting van de blitters, omdat de meeste belangstelling voor deze nieuwe machine komt van de doelgroepen voor wie

niet de blitter, maar de grote geheugenkapaciteit de belangrijkste reden tot aankoop is.

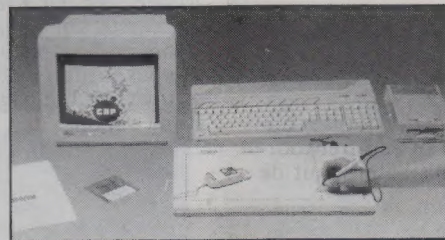
De nieuwe ROM's, waarvoor het besturingssysteem voor de Mega ST staat, werken zowel met als zonder blitter. Dat is dan ook tegelijkertijd de enige uitbreiding ten opzichte van het 'oude' TOS; verder is die alleen op een aantal punten verbeterd en zijn enkele fouten eruit gehaald. Er is dus geen sprake van een andere versie van GEM of van aanpassingen op de bestaande GEM-versie.

In de Mega ST's zullen de nieuwe ROM's standaard geleverd worden; later zullen ze ook voor de bestaande ST's beschikbaar komen. Een prijs is nog niet bekend. Atari belooft wel dat er een verlaagde introductieprijs komt.

## NIEUW BIJ PDS

PDS introduceerde op haar stand de CD-ROM van Hitachi. Via een AT kon men razendsnel een trefwoord in de bijbel terugvinden. Andere toepassingen zijn op komst en aan een verbinding met een ST wordt gewerkt.

Heel mooi was het tekentablet van de Duitse firma CRP. Naast een degelijk ontwerp viel de bijgeleverde software op. Het is mogelijk de menubalk op het grafisch tablet te projékteren, zodat er volledig buiten de muis om gewerkt kan worden. Uiteraard was de samenwerking met para-depaardje Art Director al voor elkaar.



Z-WORKS uit Gouda, te gast in de PDS stand, heeft alle benodigde konversieprogramma's ontwikkeld om teksten die op een ST geschreven zijn verder te bewerken tot professioneel zetwerk.



Ook de moeite van het vermelden waard was de introductie van een cursus die PDS in samenwerking met de academie van beeldende kunsten presenteert. Voor f 700,- kunnen ontwerpers die met de nieuwste technieken kennis willen maken een cursus van 10 ochtenden volgen.

## NIEUW BIJ SOFTPACK

Metacomco doet een greep naar de macht met een uitbreiding op haar Lattice-C compiler. Het nieuwe handboek is maar liefst 650 pagina's dik. Menu+ en de editor werden verbeterd en er zijn meer Unix-

kommando's geïmplementeerd. Bovendien zijn als extra een debug utility, make en een resource construction set toegevoegd.

Over updates gesproken: ook Modula schijnt een nieuwe versie op de markt te brengen. Tenslotte is de GEM-uitvoering van VIP ook al weer verbeterd.

TV Director moet voor de monochrome monitor worden wat Film Director voor lage resolutie is. Tijdens de beursdagen draaide er konstant een demonstratiefilmpje. Kennelijk is het mogelijk filmpjes van Film Director te converteren, want we zagen het bekende rennende jongetje. Een speciaal op monochrome toegesneden demo is in de maak.

Alladin is de Macintosh emulator die alle voorgaande emulatoren de loef zal afsteken. In ons volgende nummer hopen we door te kunnen geven of deze reclamekreet op waarheid berust. Alladin gaat f 375,- kosten. De koper moet zelf achter de benodigde Macintosh ROM's aan. Er bestaan al public domain schijfjes op Atari-formaat die gevuld zijn met Macintosh software.

## MAAK EEN MEGA-2 VAN JE ST

Met een Megabyte geheugen in je 520 ST+ of 1040 ST kun je veel doen, maar met de K-Switch en een of meer RAM-disk kun je soms nog ruimte tekort komen. En als je straks gebruik wilt gaan maken van de nieuwe Atari laserprinter kom je met een Meg ook niet zo ver.

De firma Weide-Elektronik in het West-Duitse Hilden heeft er wat op gevonden: een 2 Megabyte uitbreiding van het RAM-geheugen. De uitbreiding is geschikt voor alle huidige ST-modellen (260, 520 en 1040) en kan zonder solderen worden geïnstalleerd. Door gebruik te maken van de allernieuwste, superkompakte 1 MBit CMOS ramchips omvat de uitbreiding slechts een klein printje dat probleemloos een plaats vindt in de systeemkast. Het extra geheugen is direct bruikbaar; er is geen speciale software nodig.

Een nadeel is er wel: van de 1 Megabyte geheugen die de 520 ST+ en 1040 'van huis-uit' al hebben wordt de helft buiten gebruik gesteld, omdat de MMU (Memory Management Unit, die de interne gang van zaken regelt) niet toelaat dat dit gedeelte van het geheugen aangesproken wordt. Daarom wordt de helft van de geheugenchips buiten werking gesteld. Van de 3 Megabyte die je in theorie in huis hebt, blijft dus 2,5 Meg bruikbaar werk-geheugen over.

De uitbreiding kost f 1299,- inclusief BTW en wordt in Nederland geïmporteerd door Commedia, 1e Looiersdwarsstraat 12, Amsterdam, telefoon (0 20) 23 17 40.

Weide-Elektronik heeft nog een noviteit: de mathematische co-processor 68881, die berekeningen volgens opgave van de fabrikant 200 tot 700 keer sneller uitvoert dan de - toch al snelle - 68000-processor. De kracht van deze co-processor komt vooral tot uiting bij complexe floating point-berekeningen en bij CAD (computer-ondersteund ontwerpen).

Om gebruik te maken van deze super-rekenaar, die bovenop de Glue-chip kan worden geplaatst, moet de programmatuur enigszins aangepast worden. Dat wil zeggen: aan de sourcecode hoeft niets te worden veranderd, maar wel moet het programma opnieuw worden gecompileerd, waarbij enkele extra linkmodules worden toegevoegd. Deze modules komen beschikbaar voor een groot aantal com-



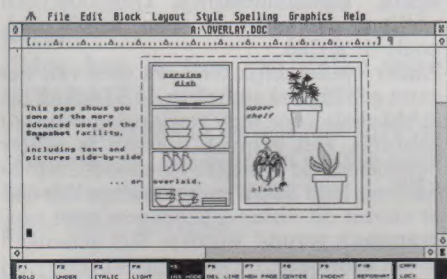
pilers, waaronder Mark Williams en Fortran. Standaard wordt een Megamax-module geleverd. Verder komt er een speciale versie van GfA Basic die de co-processor volledig ondersteunt. De importeur, Commedia, zal met distributeurs van daarvoor in aanmerking komende programma's in overleg treden over de mogelijkheid de originele programmaschijf te ruilen tegen een versie die gebruik maakt van de 68881. De co-processor kost f 1099,- inclusief BTW.

## TEKSTVERWERKERS

Goed nieuws voor notoire tekstverwerkers en andere letterploeters: er zijn betere tijden op komst. Na Beckertext, de sterk verbeterde opvolger van Textomat, komen er nog ten minste drie tekstverwerkers op de markt die zeer de moeite waard zijn.

Allereerst de langverwachte Nederlandse versie van 1st Word Plus. Dit programma, dat ten opzichte van 1st Word een groot aantal verbeteringen en uitbreidingen heeft ondergaan (onder meer invoegen van plaatjes, spellingcontrole, decimale tabulator, etc.), is nu volledig Nederlandstalig; inclusief een woordenlijst met 44.000 woorden. Ook het bijgeleverde programma 1st Mail (voor seriebrieven) is vertaald. 1st Word Plus wordt gedistribueerd door Atari Benelux en gaat f 199,- kosten.

Wie liever met de Engelse versie werkt: die is te krijgen via GST. Let in de winkel wel even op wat u koopt.



Op de Comdex-beurs, die begin juni in Atlanta (VS) is gehouden, zijn nog twee opmerkelijke tekstprogramma's voor de ST geïntroduceerd: Microsoft Write en WordPerfect.

Write is bekend onder de naam MS-Word voor IBM-PC en compatibles en voor Apple Macintosh, en kent daar vele enthousiaste gebruikers. Het pakket biedt ongekend veel mogelijkheden, die vooral veelschrijvers van pas zullen komen. Voor de doorsnee briefschrijver is het misschien allemaal wel teveel van het goede. Write wordt uitgebracht door Atari Benelux, die overweegt om op termijn een Nederlandstalige versie uit te brengen. Verkoopprijs: f 299,-.

Tenslotte WordPerfect, evenals Microsoft Word een goede bekende van de PC. Voor de grafisch georiënteerde personal computers - behalve de ST ook Apple Macintosh en Commodore Amiga - heeft The WordPerfect Company zijn paradepaardje opnieuw bewerkt. Volgens woordvoerder Ad Rietveld van de Nederlandse distributeur, Deltaware in Rotterdam, zijn alle functies zowel door middel van de muis als via toetsaanslagen te gebruiken; het summum van gebruiksgemak derhalve. Dokumenten die met WordPerfect op de ST zijn gemaakt kunnen zondermeer worden uitgewisseld met PC-dokument (en andersom); slechts een enkele functie zal niet compatibel zijn. Voorlopig zal in Nederland de Engelsta-

lige versie worden geleverd, maar er zijn plannen om het pakket - zoals ook met de PC-versie is gebeurd - een volledig Nederlandse bewerking te geven. In dat geval zal de gebruiker zijn 'oude' versie kosteloos kunnen omruilen voor een Nederlandse. Een definitieve prijs was bij het ter perse gaan van dit nummer helaas nog niet bekend.

## NEDERLANDSE TOS

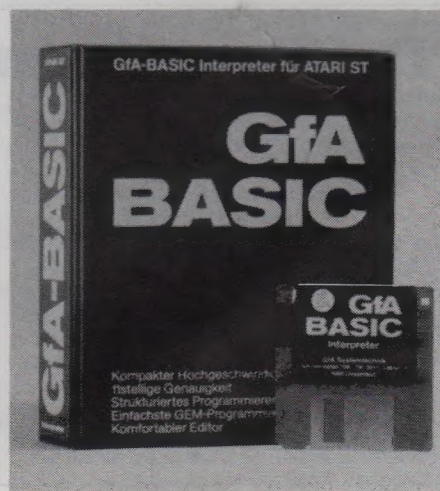
Wie wel eens met een scheef oog naar onze Oosterburen heeft gekeken omdat die door hun ST's in de moerstaal werden aangesproken, heeft niet langer reden om jaloers te zijn. Wilfred Kilwinger, software-specialist van Atari Benelux, heeft een slimme bedacht om alle schermmededelingen van TOS, het besturingssysteem, Nederlandstalig te maken. Daarvoor hoeven geen nieuwe rommetjes geplaatst te worden, en evenmin is het noodzakelijk om de volledige TOS (190 K) van schijf te laden; er moet alleen een klein programmaatje op de systeemschijf worden gezet.

Volgens Kilwinger worden de schermmededelingen elke keer bij het opstarten van het systeem van het ROM-geheugen naar het RAM-geheugen gekopieerd. Het programmaatje zorgt ervoor dat in plaats van de Engelse teksten die in de rommetjes staan gebrand, de Nederlandse teksten van de systeemschijf worden geladen. Het programmaatje, dat vooral van belang is voor de introductie van de ST op de basisscholen (maar natuurlijk ook plezierig voor degenen die moeite houden met het Engels), zal binnenkort via het public domain gratis beschikbaar worden gesteld aan iedereen die erin geïnteresseerd is.

## GfA OP ROM-MODULE

De computer aanzetten en meteen met GfA BASIC aan de slag. Dat kan met de GfA BASIC ROM-module van Weide-Elektronik, die je in de cartridgepoort aan de zijkant van de ST steekt. Door op het speciale ikoon dat op de desktop verschijnt te klikken, kom je direct in GfA terecht.

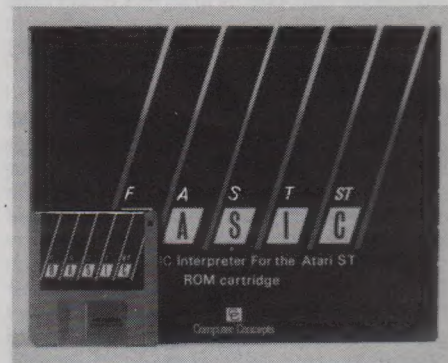
Alle mogelijkheden van de GfA-interpret (versie 2.00) zijn ingebouwd; er hoeft dus niets meer van disk geladen te worden. Bovendien neemt het BASIC geen geheugenruimte meer in beslag; dat scheelt toch weer meer dan 50 K. De prijs is iets hoger dan GfA op diskette: 259 gulden (inclusief handleiding en BTW). De module wordt gedistribueerd door Marko, Mat. Marisstraat 76, Dordrecht. Daar is ook (tegen portokosten) een GfA reference card verkrijgbaar, waarop alle komman-



do's op overzichtelijke wijze zijn gegroepeerd. Stuur een aan jezelf geadresseerde A5-envelop met f 1,50 aan extra postzegels aan bovenstaand adres en de kaart wordt per omgaande toegezonden.

## FAST BASIC-GEBRUIKERSGROEP

Leonard Kater uit Utrecht heeft het initiatief genomen tot oprichting van een FAST BASIC gebruikersgroep. Hij vindt dat FAST BASIC, dat een aantal voordelen heeft boven andere BASIC-varianten op de ST, te weinig bekend is in Nederland.



Door gezamenlijk in te kopen zou FAST BASIC met zo'n 60 gulden korting gekocht kunnen worden. Ook kunnen via de gebruikersgroep programma's, informatie, toepassingen en ervaringen worden uitgewisseld. Verder wordt gedacht aan bijvoorbeeld een eigen Bulletin Board en een nieuwsbrief. In de eerste nieuwsbrief staat een aantal aanvullingen op de handleiding van FAST BASIC.

Het lidmaatschapsgeld bedraagt voorlopig f 10,- (met recht op een tientje korting op de later vast te stellen definitieve contributie); de nieuwsbrief is ook apart verkrijgbaar voor f 5,-.

Inlichtingen bij Leonard Kater, Gaasterland 48, 3524 CB Utrecht, telefoon (030) 89 38 17.

## GEM-KOLLEKTIE VOOR DE ST

Digital Research, de ontwikkelaar van GEM, heeft verschillende softwarepakketten op de markt gebracht die optimaal gebruik maken van de mogelijkheden van dit grafische besturingssysteem. Tot nog toe waren die wel voor IBM PC's en compatibles verkrijgbaar, maar niet voor de ST - even afgezien van een aantal voorversies in het illegale kopieercircuit.

Toen Atari twee jaar geleden de ST aankondigde, was er sprake van dat deze compleet met de hele GEM-kollektie (behalve de desktop ook een tekstverwerker en een tekenprogramma) geleverd zou worden. Zover is het nooit gekomen; in plaats van deze programma's werden 1st Word en DB Master One meegeleverd.

Over de redenen waarom de GEM-kollektie niet voor de ST is uitgebracht, doen de wildste verhalen de ronde: Digital Research en Atari zouden laaiende ruzie hebben. Op AtariNet, het eigen bulletin board, liet Atari Benelux onlangs weten dat er van ruzie geen sprake is, „maar zoals gebruikelijk in business, probeert Digital Research zijn huid zo duur mogelijk te verkopen, terwijl Atari 'Power without the price' wil. De samenwerking op technisch gebied is beter dan ooit.”

Hoe het ook zij, vast staat dat de complete GEM-kollektie voor de ST alsnog op de markt gebracht wordt. ABC Software in Bruchem, de Nederlandse vertegenwoordiger van Digital Research, is al druk bezig met een volledig Nederlandse versie





van al deze programma's. De Nederlandse kollektie omvat GEM Write, GEM Paint, GEM Draw, GEM Graph, GEM Ledgers en de (al eerder uitgebrachte) eigen programma's van ABC Software, GEM Cash en GEM Viditel.

## GEM DESKTOP PUBLISHER

Ook het gloednieuwe programma GEM Desktop Publisher wordt direct in het Nederlands op de markt gebracht. Het gaat hierbij om een pakket voor de opmaak van pagina's, dat de goedkope laserprinter van Atari volledig ondersteunt.

Volgens ABC Software is GEM Desktop Publisher gespeend van grafisch vakjargon. Daardoor is het snel en eenvoudig door iedereen te gebruiken. Tegelijkertijd biedt het pakket de ervaren gebruiker een groot aantal professionele opmaakfuncties, zoals het automatisch om een illustratie zetten van tekst, variabele spatiering, invoegen van tekeningen en gescande afbeeldingen en het vastleggen van stijlen en pagina-indelingen. Bovendien past het programma zich aan bij het format van de gebruikte tekstverwerker, zodat documenten probleemloos ingelezen kunnen worden.

Een definitieve prijs van dit desktop publishing programma is nog niet bekend, maar zeker is wel dat die rond de duizend gulden zal komen te liggen.

## VIDEO GENLOCK INTERFACE

Het beeld van de ST opnemen op de video kon al via een RGB-interface; nu bestaat ook de mogelijkheid videobeelden te mengen met het computerbeeld. Dat kan door gebruik te maken van een composite video genlock interface, ontwikkeld door DPSoft in Hellevoetsluis.

De interface geeft de ST twee extra aansluitingen (video-in en video-out). Het mixen van beelden gebeurt op 'chroma-level', dat wil zeggen: je kiest een kleur (uit het palet van 512 dat de ST standaard biedt), die door de interface wordt vervangen door het videosignaal, terwijl alle

andere kleuren ondoorzichtig blijven. Op deze manier is het bijvoorbeeld ook mogelijk een stukje videobeeld in te voegen in een tekstblok. Alle bekende teken- en animatieprogramma's, zoals Art en Film Director, kunnen in combinatie met de genlock-interface worden gebruikt.

Volgens DPSoft, dat binnenkort ook een professionele RGB-uitvoering van de interface op de markt wil brengen, is de ST met de genlock interface een zeer voordelige combinatie voor iedereen die tekst, tekeningen, logo's, grafieken en animaties wil mengen met videobeelden. Daarbij wordt gedacht aan reclamebureaus, videoklubs en lokale omroepen.

De genlock-interface wordt - compleet met bijbehorende software - geleverd als inbouwset voor de ST. Prijs: f 998,- exclusief BTW.

Informatie: DPSoft, Rijksstraatweg 256, Hellevoetsluis, telefoon (0 18 83) 1 84 25.

## TWEE HARDWARE MS-DOS EMULATORS

Hij is al vaak beloofd, er is veel over gepraat, maar eindelijk is 'ie er dan: de langverwachte hardware MS-DOS emulator voor de Atari ST. Vrijwel tegelijkertijd zijn er liefst twee op de markt gebracht, elk met hun eigen bijzonderheden en kwaliteiten.

Op de CES-beurs, eind mei in Chicago (VS), heeft de firma Robtek - inderdaad, dezelfde die eerder een tergend trage software-emulator uitbracht onder de veelzeggende naam 'MS-DOZ' (zzzzzzzz...) - het kastje gepresenteerd dat de ST compatible met IBM moet maken.

De emulator zou vrijwel alle IBM-software probleemloos draaien. Dat hebben we nog niet kunnen controleren, maar de specificaties zijn veelbelovend. De emulator beschikt over een eigen microprocessor (Intel 8086), die zowel op 4,77 als op 8 Mhz kan draaien. Verder beschikt het kastje over een eigen RAM-geheugen van 512 K, dat - als de computer in de ST-stand staat - kan dienen als extra RAM-disk voor de Atari.

De emulator is standaard uitgevoerd met een Color Graphics Adaptor (CGA), die een oplossend vermogen van 640 of 320 x 200 pixels geeft; hetzelfde als de ST in medium, respectievelijk low resolution-stand. Bovendien biedt de emulator nog ruimte aan twee uitbreidingskaarten, waardoor bijvoorbeeld ook een standaard PC-harddisk gebruikt kan worden.

De Robtek-emulator is vanaf medio juni

via de meeste Atari-dealers leverbaar. De prijs bedraagt bij introductie f 799,- (inclusief BTW).

Robtek komt overigens ook met een eigen 5 1/4-inch diskdrive op de markt, die rechtstreeks op de ST kan worden aangesloten. De drive kan worden omgeschakeld tussen 40 tracks (IBM-stand: 360 K) en 80 tracks (ST-stand: 720 K), zodat je voortaan back-ups kunt maken op goedkope floppies. Deze drive gaat f 499,- kosten.

Ook de firma World Trade Channel, bekend als importeur van het Rhotron bussysteem voor de ST, heeft een eigen MS-DOS emulator ontwikkeld. Deze kan bij de meeste ST-modellen (520 ST, 520 ST+ en 1040 STF) in systeemkast van de ST worden ingebouwd; alleen bij de 260 ST wordt de elektronika in een eigen kastje ondergebracht.

De emulator kan onafhankelijk van het bussysteem worden ingezet. Als diskdrive kan de SF 354 of SF 314 gebruikt worden (als de software op 3 1/2-inch floppies beschikbaar is) of een aan de ST aangepaste 5 1/4-inch drive.

Het bijzondere van de WTC-vinding is het feit dat die met maar liefst 1 Megabyte geheugen wordt geleverd. Aangezien MS-DOS daarvan slechts 640 K RAM ondersteunt, kan de resterende 384 K worden gebruikt als RAM-disk en/of printerspooler. Als processor wordt de Intel 8086 gebruikt, die op een kloksnelheid van 4 Mhz draait. Een hogere verwerkingssnelheid (8 Mhz) kan worden bereikt door de emulator te combineren met het Rhotron bussysteem. Tegen meerprijs kan ook een snellere processor worden ingebouwd, zoals de 80286.

Ander opmerkelijk feit: een deel van het extra geheugen kan ook in de ST-stand als RAM-disk voor gegevensopslag worden gebruikt. Het gebruik hiervan is echter beperkt tot de 1 Megabyte-modellen (520 ST+ en 1040 STF), en bovendien kunnen er vanuit dit virtueel geheugen geen programma's gerund worden. Het geheugen is wel gebufferd: de gegevens worden - bij een uitgeschakelde computer - een jaar lang vast gehouden. Daardoor is het mogelijk om een aantal programma's en accessoires bij het opstarten van de computer vanuit de RAM-disk razendsnel naar het werkgeheugen van de ST te kopiëren.

De MS-DOS emulator is medio juni leverbaar en gaat in de standaard-uitvoering f 900,- (inclusief BTW) kosten.

Informatie: World Trade Channel in Kerkrade, telefoon (0 45) 42 47 77.



Computers And Music Systems

**Advisering in gebruik en toepassing  
van computers in de muziek**

## De ST - speciaalzaak in Utrecht

**official dealer voor:**

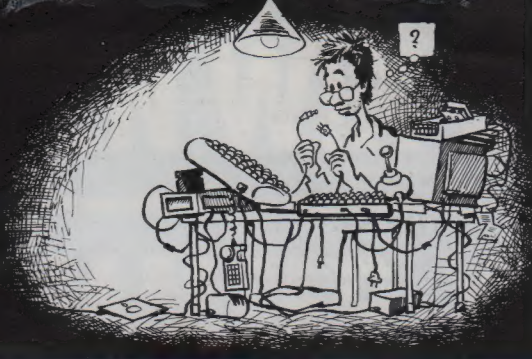
**Akai-professional  
Atari-ST  
Fuji-diskettes  
Philips-monitoren  
Nakajima-printers  
Yamaha**

**Voorstraat 22, Utrecht tel. 030 - 314250**



# GfA

## VRAAGBAAK



GfA BASIC is dermate populair dat zelfs sommige verstokte C- en machinetaalprogrammeurs zich er toe bekeren. Het leek de redactie dan ook een goed idee om een rubriek te beginnen waarin gebruikers tips kunnen uitwisselen.

Heeft u een truukje ontdekt dat volgens u de moeite waard is?

Stuur het op. Elke tip die het plaatsen waard is, wordt gehonoreerd met een START-diskette.

GfA VRAAGBAAK is ook de rubriek waarin u vragen kunt stellen.

"Ik heb de bijgesloten routine geschreven. De GfA compiler accepteert hem echter niet. Wie weet een oplossing?"

"Hoe laat je de cursor knippen?"

"Hoe kun je zien van welke drive GfA-BASIC geladen werd?"

Onze 'vraagbaak in levende lijve' is Heino Kaal, die niet zal rusten voor hij de oplossingen voor uw problemen heeft gevonden.

### ● PERFЕКTE INVOER

Omdat er de eerste aflevering van een nieuwe rubriek natuurlijk nog geen post te beantwoorden is, heb ik zelf maar het voortouw genomen. U krijgt een gebruiksvriendelijke invoerroutine kado.

Misschien denkt u: "Waar heb ik een invoerroutine voor nodig; GfA heeft toch diverse invoer-functies?" In dat geval zal ik u even uit de droom helpen. Laten we om te beginnen 'Input' eens bekijken:

Input "Geef uw naam...": ",Nm\$

Als u dit in uw programma gebruikt, komt er een vraagteken op het scherm. In veel gevallen is dat nogal hinderlijk. In ons voorbeeld is "Geef uw naam" tenslotte helemaal geen vraag!

Gelukkig is dit op te lossen door de puntkomma te vervangen door een komma:

Input "Geef uw naam ...": ",Nm\$

Opgelost? Dat vraagteken wel, maar stel dat u in een ander geval een invoer van exact tien numerieke tekens (cijfers) wilt. Hoe bepaal je dan de grens van tien en de controle op cijfers? En hoe gaat het als je komma's of stuurtekens als invoer wilt gebruiken? Het gewone 'Input' is voor dit soort toepassingen volstrekt onvoldoende.

Er is een instructie die meer mogelijkheden biedt: Form Input. Tik listing 1 in en laten we de routine eens gebruiken. Voer bijvoorbeeld uw naam in en druk op 'return'. U ziet dat de invoer niet geaccepteerd wordt. Dat komt door de 'Val'

instructie achter 'Until'. Deze routine werkt alleen getallen.

Tik nu eens de cijfers 12345678901. Van de eerste '1' tot de '0' gaat alles goed. De tweede '1' wordt echter niet geslikt. Dat komt omdat we de lengte gedefinieerd hebben met de opdracht 'Form Input 10,Iv\$'.

Druk op 'return' en volg de scherm-instructie op. Nu is er van uw input alleen nog maar 34567890 te zien en Iv\$ is leeg. De '1' en de '2' zijn verdwenen. Dat is lastig als u iets wilt verbeteren. U moet dan toch kunnen zien wat uw oude invoer geweest is.

Tik achter "Verbeter uw input...:" de cijfers 12345. U zult zien dat er, elke keer als de cursor naar rechts gaat, een oud cijfer wordt weggehaald. Dat betekent dat er bij het verbeteren van fouten niet van de vorige invoer geprofiteerd kan worden. Kortom: ook 'Form Input' laat te wensen over.

Gelukkig heeft Frank Ostrowski (de ontwerper van GfA-BASIC en -compiler) nog meer voor ons in petto. Nadat u op een toets heeft gedrukt, zijn we toe aan deze laatste instructie:

Form Input 10 As Iv\$

En kijk: zonder dat we een aparte print-instructie hebben gebruikt, komt Iv\$ op uw scherm met de cursor op de eerste positie. Toen ik dit kommando voor de eerste maal zag, was ik in de zevende hemel, maar dat duurde maar even. Wat is er nu weer mis met deze in opzet zeer nuttige instructie?

Tik maar eens 678901, dan ziet u wat ik bedoel. Over oude karakters heen tikken is onmogelijk. Je kunt de insert-toets in blijven drukken, maar het helpt niets. Insert kan niet uitgeschakeld worden. Er moet na tien karakters gewist worden voor het kommando weer karakters accepteert.

### ● INSTELBARE PARAMETERS

Het zou prettig zijn als de volmaakte invoerroutine standaard in GfA-BASIC zou zitten, maar er moet natuurlijk voor ons ook nog wat te programmeren overblijven.

In de hoop u met de voorbeelden van het nut van de routine overtuigd te hebben, vraag ik uw aandacht voor listing 2.

Het programma begint met een paar variabelen te definiëren. De instructie die daarop volgt is degene die u voortaan in uw eigen programma's zult gebruiken om de invoerroutine aan te roepen:

@ Invoer(20,14,25,1,0,0,0,1)

Het '@' (apestaartje) is een afkorting van 'Gosub'. De getallen tussen haken worden meegenomen naar de Procedure Invoer(). Zoek hem eens op in de listing. U ziet daar variabelen tussen de haken staan. Deze variabelen krijgen de waarden die we bij de aanroep (de 'Gosub') hebben opgegeven:

De variabele 'Kolom' krijgt dus de waarde '20' en 'Regel' de waarde '14'. Hierdoor zet de routine de cursor op regel 14, kolom 20.

U kunt dus zelf bepalen WAAR de invoerroutine zijn werk moet verrichten.

De derde variabele, I-lengte, wordt gevuld met het getal '25'. Hierdoor accepteert de invoerroutine "slechts" 25 toetsaanslagen. De maximale lengte die opgegeven mag worden is 32767 karakters. Wie daar nog niet genoeg aan heeft mag het zeggen.

De resterende variabelen zijn 'vlaggen'. Hierbij geldt: '0' is UIT en '1' is AAN.

De eerste vlag, I-leeg, staat in het voorbeeld AAN (1). Dit betekent dat de gebruiker verplicht is om iets in te voeren.

Zou de vlag UIT (0) staan, dan accepteert de routine een lege invoer (alleen op 'return' drukken).

De tweede vlag, I-cijfer, staat in het voorbeeld UIT (0). Alle tekens van het toetsenbord mogen daardoor ingevoerd worden. Run het programma maar eens met een gehesen vlag (1). U zult zien dat de routine plots alleen nog maar van cijfers geïndiend is.

De derde vlag, I-upper, bepaalt of de invoer met grote en kleine letters of alleen met grote letters op het scherm gezet zal worden.

I-upper=1: invoer met grote of kleine letters.

I-upper=0: invoer alleen maar met grote letters.

De truuk die het programma hier toepast is erg waardevol. Stel dat de gebruiker van uw programma de 'J' toets in moet drukken. Dit kunt u op verschillende manieren oplossen, maar de kortste methode is:

Repeat

Wacht\$=Upper\$(Inkey\$)

Until Wacht\$="J" or Wacht\$="N"

De gebruiker hoeft hierdoor zelf niet op te letten of de 'capslock' toets op grote of op kleine letters staat.

Op de zevende positie vinden we I-spatie, de vierde vlag. Hier geldt de volgende afspraak:

I-spatie=0: de lengte van de invoer wordt bepaald door het aantal ingetikte karakters. De gebruiker kan dus onder het maximum van I-lengte blijven.

I-spatie=1: de lengte van de invoer is altijd gelijk aan het maximum. Het programma voegt eventueel spaties toe om dit te bereiken.

De laatste vlag (variabele Esc-toets) bepaalt of de Esc-toets gebruikt kan worden.

Esc-toets=1: als u tijdens de invoer op de Esc toets drukt, wordt de routine onderbroken, de string leeg gemaakt en de variabele Ctrl op '1' gezet. Bij terugkeer uit de invoerroutine kan dus onderzocht worden hoe de gebruiker hem verlaten heeft.

Esc-toets=0: de Esc-toets is niet te gebruiken.

De routine is niet alleen erg gemakkelijk voor de programmeur, maar ook voor de gebruiker. Die kan 'backspace', 'undo', 'insert' en de cursorpijltjes gebruiken om zijn ingevoerde tekst te corrigeren.

Ik hoop dat u evenveel plezier van dit hulpprogramma heeft als ik.

Wie tot slot nog antwoord op de vragen uit de inleiding zou willen hebben: u zet de "cursorknipper" uit met 'Void Xbios (21,3)'.

'X=Gemdos(25)' bepaalt van welke drive u GfA-BASIC heeft geladen. Als X na de aanroep '0' bevat, was het drive A, '1' betekent drive B, enz.

Tot de volgende keer. En vergeet uw tips niet op te sturen.

U kunt het volgende adres gebruiken:

START REDAKTIE

GfA-rubriek

M. Gijzenburg 14

2907 HG Capelle a/d IJssel



## LISTING 1

```

Repeat
  Print At(2,2); "Invoer numeriek ....: ";
  Form Input 10, Iv$
Until Val(Iv$) <> 0
@Toets(2,4)
Repeat
  Print At(2,2); "Verbeter uw input ..: " =>
  Iv$; At(2,2);
  Form Input 10, Iv$
Until Val(Iv$) <> 0
@Toets(2,4)
Repeat
  Print At(2,4); "Probeer het nog eens: ";
  Form Input 10 As Iv$
Until Val(Iv$) <> 0
Print At(2,6); "Laatste invoer is ...: "; Iv$
@Toets(2,8)
Edit
Procedure Toets(Ko, Re)
  Print At(Ko, Re); "Druk op een toets"
  Void Inp(2)
  Print At(Ko, Re); Space$(17)
Return

```

## LISTING 2

```

Esc$=Chr$(27)
Ca$=Esc$+"e" ! cursor aan.
Cu$=Esc$+"c" ! uit.
Ra$=Esc$+"p" ! rvs aan.
Ru$=Esc$+"q" ! uit.
Print At(6,14); "Test invoer : "
@Invoer(19,14,25,1,0,0,0,1)
Print At(1,1); "Input = "; Invoer$; "L =>
engte input = "; Len(Invoer$)

' Kolom = Kolom waar de cursor op =>
' een regel komt te staan.
' Regel = Regel waar de cursor ko =>
' mt te staan.
' I_lengte = Input lengte
' I_leeg = 0 Een lege Input is wel =>
' toegestaan.
' = 1 lege Input is nie =>
' t toegestaan.
' I_cijfer = 0 Input alfa-nummeriek.
' = 1 nummeriek.
' I_upper = 0 Input groot/klein.
' = 1 grote letters.
' I_spatie = 0 Input niet opgevuld =>
' et spaties.
' = 1 wel
' Esc_toets = 0 Geen actie op Esc toets.
' = 1 Wel actie.

' Het invoer resultaat komt in de var =>
' iable Invoer$ te staan.

Procedure Invoer(Kolom, Regel, I_lengte =>
, I_leeg, I_cijfer, I_upper, I_spatie, Esc =>
_toets)
  Hide$
  Local Hulp$, C_toets, Key, Insert, Re_t =>
  urn, I_tot, C_pos

  ' Hulp$ = begrijpelijk.
  ' C_toets = cursor toetsen (de 4 =>
  pijltjes onder Help en Undo)
  ' Key = controle of de inser =>
  t, backspace, of een cursor toets is =>
  ingetoetst.
  ' Insert = wel of niet actief.
  ' Re_turn = de ingedrukte toets i =>
  s de return toets.
  ' I_tot = Invoer totaal (einde =>
  (I_lengte) bereikt?).
  ' C_pos = positie waar de curso =>
  r zich bevindt.

```

```

Clr Invoer$, Ctrl
C_pos=1
Invoer$=Space$(I_lengte)
Repeat
  Print At(Kolom+C_pos, Regel); Ca$;
  Repeat
    Hulp$=Inkey$
    Until Len(Hulp$)
    Print Cu$;
    Key=Asc(Mid$(Hulp$, 2, 1))
    If Esc_toets And Hulp$=Esc$
      Ctrl=1
      Clr Invoer$
      Goto I_einde
    Endif
    If I_tot
      If Asc(Hulp$)=127 And C_pos<=I_ =>
      tot
        ! delete toets
        Dec I_tot
        Mid$(Invoer$, C_pos)=Mid$(Invo =>
        er$, C_pos+1)+ "
        Print At(Kolom+C_pos, Regel); M =>
        id$(Invoer$, C_pos);
        Endif
        If Key=82 And Insert=0 =>
        ! Insert actief.
        Insert=1
        Clr Key
        Endif
        If Key=82 And Insert=1 =>
        ! Insert niet actief.
        Clr Insert
        Endif
        If Key=97 =>
        ! Undo toets
        Print At(Kolom+1, Regel); Space =>
        $(I_lengte);
        Invoer$=Space$(I_lengte)
        Clr I_tot, Insert
        C_pos=1
        Endif
        If Key=72 Or Key=75 Or Key=77 O =>
        r Key=80
          C_toets=1
          If Key=75 =>
          ! cursor links
            Dec C_pos
            If C_pos<1
              C_pos=1
            Endif
            Endif
            If Key=77 And C_pos<=I_tot =>
            ! crsr rechts
              Inc C_pos
              If C_pos>I_lengte Or C_pos =>
              I_tot
                @Inv_sp
                Endif
                Endif
                If Key=72 =>
                ! crsr omhoog
                  C_pos=1
                  Endif
                  If Key=80 =>
                  ! crsr omlaag
                    @Inv_sp
                    Clr C_toets
                    Endif
                    Endif
                    If (Hulp$<" " Or Hulp$>Chr$(122)) =>
                    And Not (Hulp$=Chr$(8) Or Hulp$=Chr$( =>
                    (13))
                      Goto I_einde
                    Endif
                    If Hulp$=Chr$(8) Or Hulp$=Chr$(13 =>
                    ) ! Backspace of return.
                      If Hulp$=Chr$(8) And C_pos>1 =>
                      ! al iets ingevoerd ?
                        If C_toets
                          Dec C_pos

```

```

Mid$(Invoer$, C_pos)=Mid$(In =>
voer$, C_pos+1)+ "
Print At(Kolom+C_pos, Regel) =>
; Mid$(Invoer$, C_pos);
Else
  If C_pos<>I_tot
    Dec C_pos
  Endif
  Mid$(Invoer$, C_pos, C_pos)=" "
  Print At(Kolom+C_pos, Regel) =>
  ; " ";
  Endif
  Dec I_tot
Else
  If Hulp$=Chr$(13)
    If I_tot=0 And I_leeg=1 =>
    ! Geen lege input toegestaan.
      Out 2, 7
    Else
      Re_turn=1 =>
    ! Return toets
    Endif
    Endif
    Endif
    Goto I_einde
  Else
    If I_upper
      Hulp$=Upper$(Hulp$)
    Endif
    If C_pos<I_lengte And Len(Hulp$)
      If I_cijfer And Instr("012345 =>
      6789-. ", Hulp$)=0
        Out 2, 7
        Goto I_einde
      Endif
      Endif
      Endif
      If Insert And C_pos<I_tot =>
      ! insert is actief
        Mid$(Invoer$, C_pos)=Hulp$+Mid$( =>
        Invoer$, C_pos, I_lengte)
        Print At(Kolom+C_pos, Regel); Mid =>
        $(Invoer$, C_pos);
      Else
        Mid$(Invoer$, C_pos, C_pos)=Hulp$
        Print At(Kolom+C_pos, Regel); Mid =>
        $(Invoer$, C_pos, 1);
      Endif
      Inc C_pos
      Inc I_tot
      If I_tot>I_lengte
        I_tot=I_lengte
      Endif
      If C_pos>I_lengte
        C_pos=I_tot
      Endif
      I_einde:
      Until Re_turn Or Ctrl
      If I_spatie=0
        While Right$(Invoer$, 1)=" " =>
        ! Spatie's achter input weghalen.
          Invoer$=Left$(Invoer$, Len(Invoe =>
          r$)-1)
        Wend
      Endif
      Show$
      Return
    Procedure Inv_sp
      J=Len(Invoer$)
      Repeat
        If Instr(J, Invoer$, " ")
          Dec J
        Endif
        Until Mid$(Invoer$, J, 1)<>" " Or J=0
        If J=I_lengte
          C_pos=J
        Else
          C_pos=J+1
        Endif
        I_tot=C_pos-1
      Return

```

\*\*\*\*\*



Official PC-Dealers \*\*\*\*\*



**COMPUdress** bv

Postbus 6 - 3470 CA Kamerik (Utr. bij Woerden)  
Tel. 03481-2124 - Burg de Kockstraat 11 - Kamerik

**hobbelink**  
**computers**

Tuindorpstraat 2-8 Hengelo Tel. 427275

'Voor wie zich niet kan veroorloven  
om achter te lopen ...'

**Commedia**

Open wo t/m za 10-6, do tot 9 uur

Eerste Looiersdwarsstraat 12  
1016 VM Amsterdam Tel. 020 - 23 17 40



# DOE MEE EN WIN IN DE ATARI PROGRAMMEER-WEDSTRIJD!

Steeds meer Atari computers worden op basisscholen geplaatst. Daardoor is er een behoefte ontstaan aan educatieve software. Atari wil door middel van deze wedstrijd een aantal goede educatie programma's voor de ST computers beschikbaar krijgen. Doe dus mee en win één van de schitterende prijzen.

**1<sup>e</sup> PRIJS: MEGA ST4 met SC1224 Kleur-Monitor**

**2<sup>e</sup> PRIJS: MEGA ST2 met SC1224 Kleur-Monitor**

**3<sup>e</sup> PRIJS: 1040STf met SC1224 Kleur-Monitor**

**EXTRA PRIJZEN VOOR  
VERLOTING ONDER CLUBS:**

**1<sup>e</sup> PRIJS:** Laser-Printer SLM

**2<sup>e</sup> PRIJS:** Harddisk SH204

**3<sup>e</sup> PRIJS:** Kleuren-Monitor SC1224

**EXTRA PRIJZEN VOOR  
VERLOTING ONDER SCHOLEN:**

**1<sup>e</sup> PRIJS:** 1040STf met SC1224  
Kleur-Monitor

**2<sup>e</sup> PRIJS:** 1040STf met SM125  
Z/W-Monitor

**3<sup>e</sup> PRIJS:** 520STm met SM125  
Z/W-Monitor

## ALTIJD PRIJS!

Elke deelnemer wint in ieder geval een body-warmer, een T-shirt of een sporttas.

**TOTAAL  
f 20.000,-  
AAN PRIJZEN!**



### ATARI 520 ST™

Met zijn prestaties heeft ATARI de richting bepaald. Ook wat prijs betreft. Professionele computer-mogelijkheden voor ongewoon lage prijzen, dat kan alleen door de modernste technologie te gebruiken. ATARI, geavanceerde computertechnologie van vandaag, voor mensen die met meer prestaties meer willen bereiken. ATARI 520 ST™ richtprijs f 699,-

**ATARI®**  
... maakt computers betaalbaar.



Vaker en vaker worden de ATARI ST-Computers op scholen ingezet. In Utrecht bijvoorbeeld worden op alle 120 basisscholen de ATARI 1040 STf-kleurenssets geplaatst. De Inspecteur voor het Onderwijs heeft in overleg met vele deskundigen en instanties, deze keus bepaald. Met name het grote geheugen, de grafische beeldschermopbouw, de lage prijs maar vooral de gebruiksvriendelijkheid

door de muisbesturing, hebben de doorslag gegeven.

De Stichting Nederlandse Schoolradio zal in samenwerking met de VARA en AVRO via radio- en televisieprogramma's de ATARI ST als schoolcomputer begeleiden.

De behoefte naar educatieve software ontstaat. Maak een kans op één van de drie onderstaande hoofdprijzen.

## WEDSTRIJDVOORWAARDEN

- Alleen programma's met een edukatieve waarde kunnen meedingen in de wedstrijd.
- Alleen programma's welke door ATARI worden geaccepteerd dingen mee in de wedstrijd.
- Programma's welke door ATARI niet kunnen worden geaccepteerd, zullen begeleid met een standaardbrief aan de inzender worden geretourneerd.
- Alleen programma's op een diskette mogen worden ingezonden, vóór de sluitingsdatum van 1 september 1987 aan:  
**ATARI (Benelux) BV**  
**Postbus 70**  
**4130 EB Vianen**
- Alle rechten op de hoofdprijs-programma's alsmede de winnende school- en club-programma's worden eigendom van ATARI.
- Programma's ontwikkeld in clubverband door regionale clubs, b.v. de Stichting ATARI Gebruikers of van de HCC-ATARI Gebruikers
- Groep maken d.m.v. een extra verloting kans op één van de speciale Club-Prijzen.
- Programma's ontwikkelt in samenwerking met, of door een school maken d.m.v. een extra verloting kans op één van de speciale School-Prijzen.
- Programma's worden door een deskundige jury beoordeeld op de volgende eigenschappen:
  - Edukatieve waarde
  - Bedieningsgemak (Muis)
  - Originaliteit
  - Uitvoering
  - Documentatie
- ATARI werknemers en hun familieleden zijn uitgesloten van deelname.
- Het besluit van de jury is definitief en over de uitslag kan niet gecorrespondeerd worden.
- Deze wedstrijd is nietig indien belemmerd door wettelijke maatregelen.

## DE PRIJZEN

### HOOFDPRIJZEN:

- 1e PRIJS:** MEGA ST4 met SC1224 Kleur-Monitor
- 2e PRIJS:** MEGA ST2 met SC1224 Kleur-Monitor
- 3e PRIJS:** 1040STf met SC1224 Kleur-Monitor

### CLUBPRIJZEN:

- 1e PRIJS:** Laser-Printer
- 2e PRIJS:** Harddisk SH204
- 3e PRIJS:** Kleuren-Monitor SC1224

### SCHOOLPRIJZEN:

- 1e PRIJS:** 1040STf met SC1224 Kleur-Monitor
- 2e PRIJS:** 1040STf met SM125 Z/W-Monitor
- 3e PRIJS:** 520STm met SM125 Z/W-Monitor

**TOTAAL f20.000,- AAN PRIJZEN!**



# 1ST AID

**1ST-Word is de meest gebruikte tekstverwerker voor de ST. Bijna alle teksten voor START worden als .DOC-bestanden aangeleverd. Reikhalzend werd dan ook uitgezien naar de opvolger: 1ST-Wordplus. Wat voor nieuwe opties zouden aan het pakket toegevoegd worden?**

**Nu de (semi-) definitieve versies in omloop zijn gekomen, is het antwoord op die vraag bekend. Maar naast de vele mogelijkheden zijn ook de beperkingen zichtbaar geworden. Zo is het bijvoorbeeld niet mogelijk om een tekst in twee kolommen af te drukken. Ook een optie om automatisch een trefwoordenregister te maken, is pijnlijk afwezig. Onze vaste programmeur Wim de Winter besloot zich hier niet bij neer te leggen. 1ST-Aid, het resultaat van zijn werkzaamheden, vult de lacunes op en heeft zelfs nog twee extra mogelijkheden.**

## GEBRUIK

Als u het programma ingevoerd en gestart heeft, verschijnt het volgende menu:

1. ASCII-- > 1ST-Word
2. 1ST-Word-- > ASCII
3. 1ST-Word in twee kolommen
4. 1ST-Word indexen

Elk ander getal breekt het programma af.

Laten we de opties van 1ST-Aid eens nader bekijken.

## 1. KONVERSIE NAAR 1ST-WORD

Wie meerdere tekstverwerkers in gebruik heeft, zal tekstbestanden graag uit willen kunnen wisselen.

1ST-Word kan een ASCII-file weliswaar lezen, maar niet formatteren. Voor het einde van elke regel verwacht het programma namelijk een stuurcode, die het geregeleinde onderscheidt van het einde van een alinea. Iets dergelijks geldt voor de spaties.

1ST-Aid komt u met zijn eerste optie te hulp. Het programma vraagt naar de naam van een ASCII-tekstbestand en zet dat om naar een bestand in 1ST-Word formaat met de extensie .DOC.

In de praktijk zal blijken, dat 1ST-Aid vaak teveel alinea's samenvoegt. U zult dus met de hand aanpassingen moeten maken. Alle mogelijkheden van 1ST-Word staan nu echter ter beschikking om de tekst het juiste aanzien te geven. Voor permanente alineascheiding in 1ST-Word: zet de cursor op de eerste letter van de nieuwe alinea. Druk daarna op <backspace> tot er geen spaties meer voor de eerste letter staan en geef een <return>. Blijven er wel een of meer spaties staan, dan zullen de twee alinea's bij de eerste reformat (F10) weer worden samengevoegd.

## 2. KONVERSIE NAAR ASCII

Nu de andere kant van het verhaal. 1ST-Word kan teksten in ASCII-formaat weg-

schrijven, maar doet dit door alle codes die niet de standaard volgen (alle karakters van 128 tot 255) gewoon weg te laten.

Alle ã-s, è-s, ü-s en ø-s worden tijdens dit proces verwijderd, wat vooral Duitse en Franse teksten onleesbaar maakt. Maar ook het Nederlands blijft niet ongeschonden. Wat te denken van 'coëfficiënt' (coëfficiënt)?

Ook schijnt het soms nodig te zijn om regels niet met <CR><LF> af te sluiten, maar met alleen <CR>.

Optie twee biedt een oplossing voor deze problemen. 1ST-Aid vraagt naar een 1ST-Word .DOC-bestand en zal dat omzetten naar een bestand met alleen ASCII-karakters.

Allereerst wordt echter de konversietabel getoond, met de mogelijkheid deze te wijzigen. In het mini-venster staan in de linkerkolom de tekens die in het DOC-bestand voor zouden kunnen komen en in de rechterkolom de vertalingen. Het apestaartje 'e' houdt in dat het teken niet wordt vertaald, maar overgeslagen (behalve natuurlijk bij het apestaartje zelf). Door op een teken in de linkerkolom te klikken, kan een nieuwe vertaling (maximaal zes tekens) worden opgegeven. Antwoordt u hier slechts met <return>, dan verandert de vertaling in 'e'.

Door de op de pijltjes te klikken, komt een ander deel van de tekenset in beeld. Als u tevreden over de konversietabel bent, kunt u hem save en later weer gebruiken. De tabel wordt weggeschreven onder de naam 1ST-AID.TRN. Met een klik op 'EXIT' begint de vertaling.

## 3. TWEE KOLOMMEN AFDRUK

Dan de mogelijkheid om een tekst in twee kolommen af te drukken. Op het moment dat u de tekst in 1ST-Word schrijft, moet u al rekening houden met de kolombreedte. Optimaal is 0.9 maal de halve regellengte van de printer. Voor een standaard 80 kolomsprinter is dat 35 of 36 tekens.

Wie een oud bestand in twee kolommen wil afdrukken, heeft het echter niet moeilijk. Inladen, de "liniaal" korter maken en een tijdje op <F10> drukken volstaat. Tijdens de konversie blijven alle tekststijlen (vet, kursief ...) behouden. Met harde en voorwaardelijke paginascheiding wordt rekening gehouden en ook plaatjes blijven mogelijk. Alleen in voetnoten is niet voorzien.

Het programma vraagt om een 1ST-Word DOC-bestand en maakt onder dezelfde naam een bestand aan met de extensie .CLM.

Vervolgens worden de printerbreedte en de tussenscheidingsbreedte gevraagd.

Antwoord met <return> voor de standaardwaardes. Na wat berekeningen (die grafisch op het scherm worden ondersteund), zal het CLM-bestand worden aangemaakt. U kunt dit hierna via 1ST-Word(+) afdrukken. Zet hierbij de linker kantlijn wat kleiner.

## 4. TREFWOORDENREGISTER

De woorden die u in het trefwoordenregister wilt opnemen (de entries), moet u in de tekst in de lichte stijl zetten.

1ST-Aid stelt daarna in twee slagen de index samen. De eerste slag verzamelt en alfabetiseert alle entries; de tweede slag zoekt alle plaatsen waar een entry voorkomt en voegt het bladzijdenummer aan de index toe.

## AANWIJZINGEN BIJ DE LISTING

*Om ruimte te winnen is de listing in kolommen afgedrukt. Wanneer u in de listing het symbool => tegenkomt, is sprake van een afgebroken regel. U moet het symbool => zelf niet intikken en ook niet op RETURN drukken; maar gewoon verder gaan met de volgende regel.*

Het is voldoende om een entry eenmaal licht te markeren. 1ST-Aid kijkt in de tweede slag naar het woord en niet langer naar de stijl.

Het markeren hoeft overigens niet per se de eerste maal dat het woord in de tekst voorkomt te gebeuren.

Iedere entry kan meer dan een woord lang zijn, maar mag niet meer dan 25 tekens bevatten (inclusief spaties). De entry moet op een regel staan. Het indexen negeert de spaties en zoekt wel over geregeleindes, ook al is een entry afgebroken (hyphenation in 1ST-Word-Plus).

Het programma vraagt of de entry-markering verwijderd moet worden. Default-waarde is hier 'nee' (return). Toetst u 'j' of 'y' in, dan wordt in het origineel ieder woord weer in zwart weergegeven. Dit is de enige optie in het hele pakket die de originele file verandert!

## DUBBELE ST

Bij het ontwikkelen van dit pakket heb ik intensief gebruik gemaakt van de process-switcher van Kuma (K-SWITCH). Zoals u misschien weet, verdeelt die de ST in twee helften. In de ene stond permanent Wordplus, in de andere op RAM-disk afwisselend GfA-Basic en een diskmonitor. Zo kon ik razendsnel met Wordplus een bestand maken en via de diskmonitor zien hoe het bestand opgeslagen werd. Op deze manier is de stuurcode van Wordplus 'gekraakt'.

De belangrijkste codes zijn:

- 1E 0A 0D einde regel
- 0A 0D einde alinea
- 1E rekbare spatie
- 1C opvulling van de 'rek'
- 1F formattersregel, bijv. liniaal of plaatje
- 0B nn voorwaardelijke paginascheiding
- 1B (80+s) stijl verandering (s= bitmasker van stijlen)

De eerste drie opties van het pakket berusten op goochelen met deze stuurcodes. Alleen de uitwerking pakt wat ingewikkeld uit. Telkens worden er een paar regels gelezen, bewerkt en naar het nieuwe bestand geschreven. De index-optie verschilt in die zin, dat er nauwelijks uit het originele bestand gekopieerd wordt.

Ten behoeve van een grote verwerkingssnelheid wordt het hele bestand in het geheugen geladen. Het verzamelen van de entries komt dan neer op het zoeken naar escapes (\$1B), gevolgd door een getal waarin het tweede bit gezet is (lichte stijl). In de tweede slag wordt elk woord vergeleken met de eerste drie letters van elke entry. Komen die overeen, dan wordt ook de rest van het woord getest. Spaties, geregeleindes en streepjes worden niet in de vergelijking betrokken.

Wim de Winter



```

*****
* Programma 1st Aid *
* Doel Hulp bij 1st-word+ *
* Auteur Wim de Winter *
*****

@Messages
@Init
Do
  Tb$=Chr$(&H9) ! tab
  Lf$=Chr$(&HA) ! line feed
  Ff$=Chr$(&HC) ! form feed
  Cr$=Chr$(&HD) ! carriage return
  Cpb$=Chr$(&HB) ! conditional p =>
age break
  Hyp$=Chr$(&H19) ! afbreekstreepje
  Esc$=Chr$(&H1B) ! escape
  Spc$=Chr$(&H1C) ! opvulspatie
  Ind$=Chr$(&H1D) ! indent
  Sft$=Chr$(&H1E) ! soft space/cr =>
If indicator
  Lay$=Chr$(&H1F) ! layout record =>
indicator

@Menu
On Keuze% Bosub Ascii_to_1stword,1s =>
tword_to_ascii,Kolommen,Index
Print #2
Close
Clear
@Messages
Loop

Procedure Kolommen
  Restyle!=False
  No_more_layout!=False
  Source_page_nr%=1
  Dest_page_nr%=1
  @Open_files(".DOC",".CLM")
  @Pagelength
  @File_header
  @Draw_page
  @Do_columns
Return

Procedure Do_columns
  Dim Line_layout_p%(2*Nr_of_lines%) =>
  Soft_page_alert%(2*Nr_of_lines%+1)
  Dim Layout_line$(100)
  Do
    If Line%<=Nr_of_lines%
      Line 161,195+Line%,189,195+Line%
    Else
      Line 286,195-Nr_of_lines%+Line% =>
      ,314,195-Nr_of_lines%+Line%
    Endif
    Inc Line%
    If Line%=Nr_of_lines%
      Inc Source_page_nr%
    Endif
    Lchr$=Left$(Line$,1)
    If Line%>2*Nr_of_lines% Or Lchr$=>
  Ff$ Or (Lchr$=Cpb$ And Asc(Mid$(Line$ =>
  ,2,1))>&H73)
    @Make_new_page
    If Lchr$=Ff$
      Let Line$=Mid$(Line$,2)
    Endif
    If Lchr$=Cpb$
      Let Line$=Mid$(Line$,3)
    Endif
    If Mid$(Line$,1,1)=Cpb$ And Asc(M =>
  id$(Line$,2,1))<=&H73
      Soft_page_alert%(Line%)=Asc(Mid =>
  $(Line$,2,1))-&H10
      Let Line$=Mid$(Line$,3)
    Endif
    Let Line_address%=Varptr(Line$)
    @Make_space
    If Instr(Line$,Hyp$)>0
      Poke Line_address%+Instr(Line$,H =>
  yp$)-1,45
    Endif
    If Restyle!
      @Restyle
    Endif
    If Instr(Line$,Esc$)>0
      @Styles
    Endif
    Pag$(Line%)=Line$
    Exit If Eof(#1)
    Repeat
      Line Input #1,Line$
      @Select_line
      Until Left$(Line$,1)<>Lay$ Or Eof =>
  (#1)
    Loop
    Inc Line%
    @Make_new_page
Return

Procedure Justify(X%)
  Local CX,IX

```

```

  CX=0
  For IX=1 To Len(Pag$(X%))
    Char$=Mid$(Pag$(X%),IX,1)
    If Char$>Chr$(31)
      Inc CX
    Else
      If Char$=Esc$
        Dec CX
      Endif
    Endif
  Next IX
  Let Pag$(X%)=Pag$(X%)+Space$(Max(0, =>
  Column_width%+Blanks%-CX))
Return

Procedure Styles
  Local IX
  IX=1
  While Instr(Line$,Esc$,IX)>0
    Inc IX
  Wend
  If Mid$(Line$,IX,1)>Chr$(&H80)
    Restyle!=True
    Style$=Mid$(Line$,IX,1)
    Let Line$=Line$+Esc$+Chr$(&H80)
  Else
    Restyle!=False
    Style$=""
  Endif
Return

Procedure Open_files(Ext1$,Ext2$)
  Path$=""
  Print At(35,2);"SOURCE FILE:"
  Repeat
    Fileselect Path$+"*"+Ext1$,"",Fil =>
  ename$
  Until Exist(Filename$) Or Filename$=""
  If Filename$<>""
    Counter.point=0
    While Instr(Filename$,"\\",Counter =>
  .point+1)
      Counter.point=Instr(Filename$," =>
  \",Counter.point+1)
    Wend
    Path$=Left$(Filename$,Counter.point)
    Nf$=Mid$(Filename$,Counter.point+1)
    If Instr(Nf$,".")>0
      Nf$=Left$(Nf$,Instr(Nf$,".")-1)
    Endif
    Print At(32,2);"DESTINATION FILE:"
    Fileselect Path$+"*"+Ext2$,Nf$+Ex =>
  t2$,Nf$
    Deffill 1,1
    Pbox 200,10,400,40
    Deffill 0,0
    If Nf$<>""
      Open "i",#1,Filename$
      Open "O",#2,Nf$
    Endif
  Endif
Return

Procedure Pagelength
  Line Input #1,Line$
  Pagelength%=Val(Mid$(Line$,2,3))
  Top_of_frm%=Val(Mid$(Line$,5,2))
  Head_margin%=Val(Mid$(Line$,7,2))
  Foot_margin%=Val(Mid$(Line$,9,2))
  Bot_of_frm%=Val(Mid$(Line$,11,2))
  Nr_of_lines%=Pagelength%-Top_of_frm =>
  %
  Z-Head_margin%-Foot_margin%-Bot_of_frm%
  If Nr_of_lines%<1
    Alert 3,Msg$(5)+Str$(Nr_of_lines% =>
  ),1,"STOP",D
    Edit
  Endif
  Dim Pag$(Nr_of_lines%*2+1)
Return

Procedure File_header
  Local A$
  Column_width%=36
  Print At(19,11);Msg$(3);
  Let Printer_width%=80
  Input Printer_width%
  If Printer_width%>0" And Printer_w =>
  idth$<""
    Let Printer_width%=Val(Printer_wi =>
  dth$)
  Endif
  Print At(35,11);": ";Space$(23);Prin =>
  ter_width%
  While Left$(Line$,1)=Lay$
    Print #2,Line$
    Line Input #1,Line$
    If Mid$(Line$,2,1)=""9"
      Column_width%=Instr(Line$,"J")- =>
  Instr(Line$,"[")+1
      Blanks%=Printer_width%-5-2*Colu =>
  an_width%
    Do
      Print At(19,12);Msg$(6);Colum =>
  n_width%

```

```

      Print At(19,13);Msg$(7);Blank =>
  sz$;": "
      Print At(59,13);2*Column_widt =>
  h%+Blanks%;": "
      Print At(19,16);"OK ";
      Input A$
      Exit If Instr("Nn",A$)=0
      Print At(19,14);Msg$(8);
      Input Blanks%
      Print At(57,14);": ";Blanks%
    Loop
    Cc%=Column_width%-1
    Cb%=Cc%+Blanks%
    Let Line$=Lay$+"9["+String$(Cb% =>
  ,".")+Chr$(&H7F)+String$(Cc%-1,".")+""]
    Endif
  Wend
Return

Procedure Init
  H$=" 1 S T - A I D"
  S$=" 1st word-plus toolkit"
  Dia=200/320
  For X1=0 To 320
    Xr=640-X1
    Y1=Dia*X1
    Yr=Dia*Xr
    Box X1,Y1,Xr,Yr
    Hidea
  Next X1
  Graphmode 3
  @Shrink(0,390,20,10,140,60,360,240)
  Box 140,60,500,300
  Graphmode 2
  Deftext 0,16,,60
  @Shrink(0,390,40,10,180,77,300,23)
  Text 180,100,H$
  Deftext 0,,18
  @Shrink(0,390,30,10,180,120,300,20)
  Text 180,140,S$
  Deftext ,,2700,4
  @Shrink(0,370,10,30,620,20,8,350)
  Text 620,20,"(c) 1987 by Wim de Win =>
  k aan Zee =>
  (NL)"
  Graphmode 3
  @Shrink(0,390,40,10,150,100,340,10)
  Box 150,110,490,100
  Deftext 1,,0,13
  Graphmode 1
  @Shrink(0,0,30,15,200,160,250,100)
  Alert 0,Msg$(1),1," OK :Author",D%
  If D%>2
    Alert 3,Msg$(2),1,"OK",D%
  Endif
  Deffill 0,0
  Pbox 140,150,500,300
Return

Procedure Shrink(P1,P2,P3,P4,P5,P6,P7 =>
  ,P8)
  Sdpoke Gintin+0,P1
  Sdpoke Gintin+2,P2
  Sdpoke Gintin+4,P3
  Sdpoke Gintin+6,P4
  Sdpoke Gintin+8,P5
  Sdpoke Gintin+10,P6
  Sdpoke Gintin+12,P7
  Sdpoke Gintin+14,P8
  Gmsys (73)
  Hidea
Return

Procedure Make_new_page
  Sound 1,15,1,3,10
  Sound 1,0,0
  If Not Odd(Line%)
    Pag$(Line%)=""
  Else
    Dec Line%
  Endif
  For IX=1 To (Line%)/2
    If Line%/2-(IX-1+Soft_page_alert% =>
  (IX))<0
      @Soft_page(IX)
    Endif
  Next IX
  For IX=1 To (Line%)/2
    If Line_layout_p%(IX)>0 Or Line_1 =>
  ayout_p%(IX+Line%/2)>0 Then
      @Print_layout_lines(IX,1)
      @Print_layout_lines(IX+Line%/2, =>
  Column_width%+Blanks%)
    Endif
    @Justify(IX)
    Print #2,Pag$(IX);Pag$(IX+Line%/2)
    Line 361,195+IX,389,195+IX
    Line 401,195+IX,429,195+IX
  Next IX
  Print #2,Ff$;
  Let Line%=1
  Arrayfill Line_layout_p%( ),0
  Arrayfill Soft_page_alert%( ),0
  Erase Layout_line$( )

```



```

Dim Layout_lines$(100)
Let Next_layout_lx=0
Inc Dest_page_nr%
Inc Source_page_nr%
@Draw_page
Return

Procedure Make_space
Let Line_address%=Varptr(Line$)
While Instr(Line$,Sft$)>0
Poke Line_address%+Instr(Line$,Sft =>
$)-1,32
Wend
While Instr(Line$,Spc$)>0
Poke Line_address%+Instr(Line$,Spc =>
$)-1,32
Wend
While Instr(Line$,Ind$)>0
Poke Line_address%+Instr(Line$,Ind =>
$)-1,32
Wend
Return

Procedure Restyle
Local P%,L1%
L1%=Len(Line$)
P%=1
While (Mid$(Line$,P%,1)<=&H20) =>
And Mid$(Line$,P%,1)<>Esc$ And P%<=L1%
Inc P%
Wend
If Mid$(Line$,P%,1)<>Esc$ And P%<=L1%
Let Line$=Left$(Line$,P%-1)+Esc$+ =>
Style$+Mid$(Line$,P%)
Endif
Return

Procedure Select_line
If (Left$(Line$,1)=Lay$ And Mid$(Li =>
ne$,2,1)<>"9") And Not No_more_layout!
If Line_layout_p%(Line%+1)=0
Inc Next_layout_lx
Let Line_layout_p%(Line%+1)=Nex =>
t_layout_lx
Endif
Layout_line$(Next_layout_lx)=Line$
Inc Next_layout_lx
Endif
Return

Procedure Print_layout_lines(IX,Start =>
_column%)
Local J%
If Line_layout_p%(IX)>0 Then
J%=0
Repeat
Layo$=Layout_line$(Line_layout_ =>
p%(IX)+JX)
If Mid$(Layo$,2,1)="8"
Mid$(Layo$,5,2)=Str$(Start_co =>
luan%-1)
Endif
Print #2,Layo$
Inc J%
Until Layout_line$(Line_layout_p% =>
(IX)+JX)=""
Endif
Return

Procedure Soft_page(IX)
If Nr_of_lines%-IX-1+Soft_page_ale =>
rt%(IX)>0
Let Newline%=2*(IX-1+Soft_page_al =>
ert%(IX))
For JX=LineX+1 To Newline%
Pag$(JX)=""
Next JX
Let LineX=Newline%
Else
Print "%1ST-E-NEVBPR, This shoul =>
d never be printed ???"
Endif
Return

Procedure Draw_page
Deffill 0,0
Graphmode 1
Pbox 140,150,500,300
Box 150,175,225,265
Box 160,195,190,195+Nr_of_lines%
Box 250,175,325,265
Box 285,195,315,195+Nr_of_lines%
Box 350,175,440,265
Box 360,195,390,195+Nr_of_lines%
Box 400,195,430,195+Nr_of_lines%
Text 170,225,Str$(Source_page_nr%)
Text 295,225,Str$(Source_page_nr%+1)
Text 391,225,Str$(Dest_page_nr%)
Graphmode 3
Return

@Make_table
Procedure Make_table
Translatie_tabel:

```

```

Data C,ue,e,ae,a,a,a,c,e,e,e,i,i,i, =>
Ae,A,<
Data E,e,e,o,o,o,u,u,y,o,Ue,c,e,Y,B =>
,f,<
Data a,i,o,u,n,N,a,o,?,e,e,1/2,1/4, =>
i,<,>,<
Data a,o,o,o,e,e,A,A,O,e,',',e,e,e,e,<
Data y,y,e,e,e,e,e,e,e,e,e,e,e,e,<
Data e,e,e,e,e,e,e,e,e,e,e,e,par. =>
,e,e,<
Data alfa,beta,gamma,pi,e,e,mu,tau, =>
e,theta,omega,delta,Phi,phi,e,e,<
Data e,ca.,>=<,<,<,<,<,<,<,<,<,<,< =>
e,e,e,<
Restore Translatie_tabel
Dim Trans$(255)
For IX=0 To 7
For JX=0 To 15
Read Trans$(16*IX+JX+128)
Next JX
Read Dummies$
Next IX
For IX=0 To 31
Trans$(IX)="e"
Next IX
For IX=32 To 127
Trans$(IX)=Chr$(IX)
Next IX
For IX=16 To 25
Trans$(IX)=Str$(IX-16)
Next IX
Trans$(9)=Chr$(9) !"<TAB>"
Trans$(10)=Chr$(10) !"<LF>"
Trans$(12)=Chr$(12) !"<FF>"
Trans$(13)=Chr$(13) !"<CR>"
Trans$(31)="" ! woord afgebroken
Box 140,150,500,300
Box 150,160,230,290
Line 165,160,165,290
Line 150,175,230,175
Line 150,275,230,275
Box 400,175,450,200
Box 400,215,450,240
Box 400,255,450,280
Deftext "",13
Graphmode 2
Text 154,173,Chr$(1)
Text 154,288,Chr$(2)
Text 410,193,"LOAD"
Text 410,233,"SAVE"
Text 410,273,"EXIT"
Text 235,165,"TRANSLATION CHARACTER =>
TABLE"
Graphmode 1
Deftext "",6
Deffill 0,0
IX=128
Klaar!=False
Showe
Do
Pbox 166,176,229,274
For JX=0 To 8
Text 154,185+10*JX,Chr$(IX+JX)
Text 180,185+10*JX,Trans$(IX+JX)
Next JX
Repeat
Until MouseX>0
Mouse X,Y,Z
Exit If ZX=2 Or Klaar!
If ZX>400 And ZX<450
Graphmode 3
Deffill 1,1
If YX>175 And YX<200
Pbox 400,175,450,200
@Load_trans
Pbox 400,175,450,200
Endif
If YX>215 And YX<240
Pbox 400,215,450,240
@Save_trans
Pbox 400,215,450,240
Endif
If YX>255 And YX<280
Pbox 400,255,450,280
Klaar!=True
Endif
Graphmode 1
Deffill 0,0
Endif
If XX<165 And XX>150
Pause 10
If YX<175 And YX>150
Sub IX,8
IX=Max(0,IX)
Endif
If YX<290 And YX>275
Add IX,8
IX=Min(247,IX)
Endif
If YX>175 And YX<275
JX=(YX-185)/10+1
If IX+JX=64
Print Chr$(7);
Else

```

```

Graphmode 3
Deffill 1,1
Pbox 151,187+10*JX,229,177+ =>
10*JX
Print At(35,12);Space$(7)
Print At(35,12);">";
Form Input 6,Trans$(IX+JX)
If Trans$(IX+JX)=""
Trans$(IX+JX)="e"
Endif
Pbox 151,187+10*JX,229,177+ =>
10*JX
Graphmode 1
Deffill 0,0
Endif
Endif
Endif
Loop
Pbox 140,150,500,300
Hidem
Return

Procedure Load_trans
Local IX
If Exist("1st_AID.TRN")
Open "i",#5,"1st_AID.TRN"
For IX=0 To 255
Line Input #5,Trans$(IX)
Next IX
Close #5
Else
Alert 3,"* File not found: 1 1ST_ =>
AID.TRN",1,"RETURN",D%
Endif
Return

Procedure Save_trans
Local IX
Open "O",#5,"1st_AID.TRN"
For IX=0 To 255
Print #5,Trans$(IX)
Next IX
Close #5
Return

Procedure 1stword_to_ascii
@Open_files(".DOC",".TXT")
@Make_table
Box 170,250,470,275
Print At(23,12);Msg$(18)
Deffill 1,1
While Not Eof(#1)
Nline$=""
Line Input #1,Line$
Pbox 170,250,470-300*(Lof(#1)-Loc =>
(#1))/Lof(#1),275
Hidem
@Make_space
If Not Left$(Line$,1)=Chr$(&H1F)
Repeat
CX=Asc(Line$)
C%=Trans$(CX)
If C%<>"e" Or CX=64
Nline$=Nline$+C%
Endif
If CX=&H1B Or CX=&HB
NX=3
Else
NX=2
Endif
Let Line$=Mid$(Line$,NX)
Until CX=0
Print #2,Nline$
Endif
Wend
Return

Procedure Index
@Open_files(".DOC",".NDX")
@Pagelength
Lof%=Lof(#1)
Close #1
Print At(21,12);Msg$(19);
Form Input 1,Answer$
Let Rem_flags!=(Instr("JjYy",Answer =>
$)>0)
@File_in_memory
@Collect_entries
If Rem_flags!
Bsave Filenames$,Start%,Lof%
Endif
@Sort_entries
@Make_index
Return

Procedure File_in_memory
Let Dimmer%=Lof%/150
Dim Index0$(Dimmer%),Index1$(Dimmer =>
%),Index2$(Dimmer%,2)
If (Not Fre(0)>Lof%/1000)
Alert 3,"Not enough memory space =>
avail-able. Decrease Randisksize or! =>
reboot with fewer accessories.",1,"CA =>
NCEL",D%

```



```

Else
  On Break Gosub Release
  StartZ=Gemdos(&H48,L:LoF%)
  Bload Filename$,StartZ
Endif
Return

Procedure Collect_entries
  IX=StartZ
  Nr_of_wordsZ=0
  Let LineZ=Space$(25)
  Deffill 0,0
  Pbox 140,150,500,300
  Deffill 1,1
  Box 170,250,470,275
  Print At(22,12);Msg$(15)
  Repeat
    If Peek(IX)=&H1B
      Inc IX
    If Peek(IX) And 2
      If Rem_flags!
        Poke IX,Peek(IX) And (Not 2)
      Endif
      Entry!=True
      Repeat
        Inc IX
        If Peek(IX)=&H1B
          Inc IX
          If (Peek(IX) And 2)
            Entry!=True
            If Rem_flags!
              Poke IX,Peek(IX) And =>
            (Not 2)
          Endif
        Else
          Entry!=False
        Endif
      Until (Peek(IX)>32 And Peek(I =>
  X-1)<>&H1B) Or Not Entry!
      JX=IX
      If Entry!
        Repeat
          Inc IX
          Until Peek(IX)=&H1B And Not =>
        (Peek(IX+1) And 2)
          If IX-JX>=3
            Bmove JX,Varptr(Line$),Mi =>
            n(IX-JX,25)
            Inc Nr_of_wordsZ
            If Nr_of_wordsZ>LoF%/150
              If Fre(0)<40*DimmerZ
                Alert 3,Msg$(4),1,"re =>
              cvoer!cancel",DZ
            Endif
            If DZ=2
              Edit
            Else
              Dim Copy$(Dim?(Index1 =>
              $()))
              Swap Copy$( ),Index1$( )
              Erase Index1$( )
              Erase Index2$( )
              Mul DimmerZ,1.5
              Dim Index1$(DimmerZ), =>
              Index2$(DimmerZ,1)
              Swap Copy$( ),Index1$( )
              Erase Copy$( )
            Endif
            Index1$(Nr_of_wordsZ)=Line$
            Index0$(Nr_of_wordsZ)=Spa =>
            ce$(25)
            SpcZ=0
            For IiZ=1 To 25
              C$=Mid$(Line$,IiZ,1)
              If C$=" "
                SpcZ=Instr(Index0$(Nr =>
                _of_wordsZ)," ")
              Mid$(Index0$(Nr_of_wo =>
              rdsZ),SpcZ,1)=C$
            Endif
            Next IiZ
            Index0$(Nr_of_wordsZ)=Lef =>
            t$(Index0$(Nr_of_wordsZ),SpcZ)
            Let Line$=Space$(25)
          Endif
        Endif
      Endif
      Inc IX
      XZ=(IX-StartZ)*300/LoF%+170
      If XZ>LxZ
        Pbox 170,275,XZ,250
        LxZ=XZ
      Endif
      Until IX>LoF%+StartZ
    Return

Procedure Sort_entries
  For IX=1 To Nr_of_wordsZ
    JZ=0
    Repeat
      Inc JX
      Until Index1$(IX)<=Index2$(JX,1) =>

```

```

Or Index2$(JX,1)=""
  If Not (Index1$(IX)=Index2$(JX,1))
    KZ=JX
    Hulp$=""
    Repeat
      Swap Index2$(KZ,1),Hulp$
      Inc KZ
      NnowZ=Max(NnowZ,KZ)
      Until Index2$(KZ,1)=""
      Index2$(KZ,1)=Hulp$
      KZ=25
      While Mid$(Index1$(IX),KZ,1)=" "
        Dec KZ
      Wend
      Index2$(JX,1)=Left$(Index1$(IX) =>
      ,KZ)
      Index2$(JX,2)=IndexC$(IX)
    Endif
  Next IX
  Erase Index1$( )
  Erase Index0$( )
  Nr_of_wordsZ=NnowZ
  For IX=1 To Nr_of_wordsZ
    Index2$=Index2$+Upper$(Left$(Inde =>
    x2$(IX,1,3))+Chr$(0))
  Next IX
  Return

Procedure Make_index
  IX=StartZ
  Print At(22,14);
  Input "start met pagina ";PageX
  Let LineZ=1
  Test$=Space$(3)
  Deffill 0,0
  Pbox 140,150,500,300
  Deffill 1,1
  Box 170,250,470,275
  Print At(22,12);Msg$(16)
  LxZ=0
  Repeat
    ChrZ=Peek(IX)
    If Peek(IX)=&H1F
      Repeat
        Inc IX
        Until Peek(IX)=&HD
        Inc IX
      Endif
      If ChrZ=&HD
        Inc LineZ
      Endif
      If ChrZ=&HC Or (ChrZ=&HB And Peek =>
      (IX+1)>&H73) Or LineZ>Nr_of_linesZ
        Inc PageX
        Let LineZ=1
      Endif
      If ChrZ>32 And (Peek(IX-1)<>&H1B)
        Bmove IX,Varptr(Test$),3
        Test$=Upper$(Test$)
        If Instr(Index2$,Test$)>0
          @Test_index(IX)
        Endif
        While Peek(IX+1)>32
          Inc IX
        Wend
      Endif
      Inc IX
      XZ=(IX-StartZ)*300/LoF%+170
      If XZ>LxZ
        Pbox 170,275,XZ,250
        LxZ=XZ
      Endif
      Until IX>StartZ+LoF%
    For IX=1 To Nr_of_wordsZ
      Swap Index2$(IX,1),Line$
      @Make_space
      Index2$(IX,1)=Space$(25)
      Lset Index2$(IX,1)=Line$
      Print #2,Index2$(IX,1);Index2$(IX,0)
    Next IX
    @Release
  Return

Procedure Release
  Void Gemdos(&H49,L:StartZ)
  Return

Procedure Test_index(PosZ)
  Local IX,JX,KZ,CX,LcZ
  IX=1
  Repeat
    StrposZ=Instr(Index2$,Test$)/4+1
    Partner$=Upper$(Index2$(StrposZ,1))
    Test2$=""
    Clr JX,KZ,LcZ
    Repeat
      CX=Peek(PosZ+JX)
      If CX>32 And LcZ<>&H1B
        Test2$=Test2$+Chr$(CX)
        Inc KZ
      Endif
      Inc JX
      LcZ=CX
    Until KZ>Len(Partner$)
    Test2$=Upper$(Test2$)

```

```

  If Test2$=Partner$
    If Instr(Index2$(StrposZ,0)," " =>
    +Str$(PageX))=0
    If Not (Index2$(StrposZ,0)="")
      Index2$(StrposZ,0)=Index2$( =>
      StrposZ,0)+", "
    Endif
    Index2$(StrposZ,0)=Index2$(St =>
    rposZ,0)+", " +Str$(PageX)
  Endif
  IX=Len(Index2$)
  Endif
  Add IX,4
  Until Instr(Index2$,Test$,IX)=0
  Return

Procedure Menu
  Deffill 0,0
  Graphmode 1
  Pbox 140,150,500,300
  Print At(29,11);Msg$(9)
  Print At(19,13);Msg$(10)
  Print At(19,14);Msg$(11)
  Print At(19,15);Msg$(12)
  Print At(19,16);Msg$(13)
  Print At(19,18);Msg$(14);
  Input KeuzeZ
  If KeuzeZ=>1 And KeuzeZ<=4
    Deffill 0,0
    Pbox 140,150,500,300
  Else
    Edit
  Endif
  Return

Procedure Ascii_to_1stword
  @Open_files(".TXT",".DOC")
  Box 170,250,470,275
  Deffill 1,1
  Print At(23,12);Msg$(17)
  While Not Eof(#1)
    Line Input #1,Line$
    Pbox 170,250,470-300*(LoF(#1)-Loc =>
    (#1))/LoF(#1),275
    Hide#
    If Line$=""
      Print #2
    Else
      Print #2,Sft$
    Endif
    Let Line_addressZ=Varptr(Line$)
    For IX=0 To Len(Line$)-1
      If Peek(Line_addressZ+IX)=32
        If Peek(Line_addressZ+IX-1)=&H =>
        1E Or Peek(Line_addressZ+IX)=&H9
          Poke Line_addressZ+IX,&H1C
        Else
          Poke Line_addressZ+IX,&H1E
        Endif
      Endif
    Next IX
    Print #2,Line$;
  Wend
  Return

Procedure Messages
  Dia Msg$(20)
  Msg$(1)="This program is strictly P =>
  D. and may not be sold nor!commercial =>
  ly published without!prior permission =>
  of the Author"
  Msg$(2)="Wim de Winter!Tappenbeckst =>
  k aan Zee"
  Msg$(3)="printer breedte (80):"
  Msg$(4)="Teveel entries.!Geheugento =>
  estand kritiek."
  Msg$(5)="Onwaarschijnlijke pagina! =>
  engte:"
  Msg$(6)="kolombreedte :
  "
  Msg$(7)="paginabreedte bij scheidin =>
  gsbreedte "
  Msg$(8)="geef scheidingsbreedte
  "
  Msg$(9)="OPTIES:"
  Msg$(10)="ASCII --> 1stWord -->
  1"
  Msg$(11)="1stWord --> ASCII -->
  2"
  Msg$(12)="Zet 1stWord-file in 2 kol =>
  omen 3"
  Msg$(13)="Index 1stWord-file -->
  4"
  Msg$(14)="Uw keuze
  "
  Msg$(15)="Entries worden verzameld"
  Msg$(16)="File-index wordt samenges =>
  teld "
  Msg$(17)="Fileconversie ASCII --> 1 =>
  st-Word"
  Msg$(18)="Fileconversie 1st-Word -- =>
  > ASCII"
  der index-vlaggen ui =>
  t origineel ? "
  Return

```



Atari Corporation, tot voor kort een verlieslijdende onderneming in spelletjes- en homecomputers, is sinds de introductie van de ST helemaal terug. Er worden indrukwekkende omzetten en dito winstcijfers gerealiseerd. Het concern is inmiddels uit de schulden en zit boordevol plannen voor nog meer nieuwe produkten tegen nog lagere prijzen.

Opmerkelijk daarbij is dat Atari zich steeds meer richt op de zakelijke gebruikers; een markt waar IBM en -compatibles en in mindere mate Apple Macintosh de dienst leken uit te maken.

Inmiddels is in Nederland de 15-duizendste ST verkocht; van dit aantal wordt naar schatting zo'n 30 percent zakelijk gebruikt. De nieuwe professionele Mega ST's, een supergoedkope laserprinter, een IBM-compatible PC, een (eind 1987 uit te brengen) grafisch werkstation en een nog snellere en geavanceerdere opvolger van de ST, de Atari TT, geven duidelijk aan waar Atari naar toe wil: een stevige positie in zowel de personal computermarkt als in de spelletjes- en homecomputermarkt en straks ook in de mini-computermarkt.

Atari-topman Jack Tramiel zei het op de Hannover Messe al: „We willen marktleider zijn in alle drie de segmenten.”

Reden genoeg om eens met Atari Benelux te gaan praten over de vraag hoe men in Nederland de zakelijke markt denkt te gaan veroveren.



## 'De ST is veel sneller terugverdiend dan de PC'



Wilfried de Graaf

een ST opvallend kort.” Dat is het antwoord van sales- en marketing manager Wilfried de Graaf van Atari Benelux op de vraag met welke argumenten hij denkt zakelijke gebruikers ervan te kunnen overtuigen dat ze beter af zijn met een ST dan met een IBM-PC.

„Vrijwel iedereen kan na een korte introductie uit de voeten met een ST, terwijl je voor een PC vaak speciale cursussen van een dag of wat nodig hebt om met een bepaald programma te kunnen werken. Die trainingen zijn de echte geldverslinders. Daar gaat vaak bijna evenveel geld in zitten als in de aanschaf. En dan nog hebben veel mensen moeite met het gebruik van hun computer. Het zou aardig zijn om eens te onderzoeken hoeveel PC's er bij bedrijven ongebruikt staan, omdat de mensen er niet mee overweg kunnen. Ik denk dat het schokkende cijfers zou opleveren.”

- Als de ST zichzelf veel sneller terugverdient dan een PC, waarom gaat Atari dan toch PC's maken? Daarmee ontkracht Atari zijn eigen verkoopargument voor de ST.

De Graaf: „We hebben te maken met een markt die verdeeld is in compatible en non-compatible computers. Zo'n 70 percent van de markt wil een IBM of -compatible, omdat de buurman die ook

heeft. Slechts 30 percent is geïnteresseerd in non-compatible. Ik denk dat de grafisch georiënteerde computers zoals de Atari ST, Apple Macintosh en de Commodore Amiga, wel in staat zijn om een paar procentjes van die markt af te snoepen. Ik verwacht zelfs dat MS-DOS z'n langste tijd heeft gehad, en dat alle computers van de nieuwe generatie zullen werken volgens het What You See Is What You Get-principe. Maar feit blijft dat de markt op dit moment, ondanks een aanzienlijk aantal nadelen, in meerderheid voor compatible computers kiest. En onze leus is Atari maakt computers betaalbaar, niet Atari maakt ST's betaalbaar.”

- What They Want Is What They Get?

De Graaf: „Precies. Overigens, de Atari PC is niet zomaar de zoveelste kloon op de markt. We hebben wel degelijk iets toegevoegd aan het IBM-concept, door standaard een hoge grafische resolutie in te bouwen en - evenals de ST - de computer te leveren met GEM en een muis. Maar als je voor een compatible computer kiest, betekent het ook dat je voor MS-DOS kiest, en dus voor een gebruiksonvriendelijk besturingssysteem dat niet meer informatie geeft dan A>. Het harde argument dat de terugverdientijd van een ST veel korter is dan van een PC, blijft volledig overeind.”

- De gebruiksvriendelijkheid van de ST is dus het belangrijkste verkoopargument. Maar hoe zit het met de software? Voor de PC zijn in de afgelopen jaren duizenden programma's van vaak zeer hoge kwaliteit geschreven. De ontwikkeling van zakelijke software voor de ST begint nu pas

goed op gang te komen.

De Graaf: „Er wordt vaak gezegd dat er onvoldoende professionele programma's zouden zijn voor de ST. Ik bestrijd dat. In de eerste plaats is het zo dat de ontwikkeling van programmatuur voor de PC pas na verloop van enkele jaren goed begon te lopen. Wat dat betreft is er voor de ST na anderhalf jaar al een opvallend groot aantal programmatitels beschikbaar; verhoudingsgewijs veel meer dan destijds voor de PC.

In de tweede plaats: voor de meest gebruikte zakelijke toepassingen zijn er ST-programma's die zich zondermeer kunnen meten met vergelijkbare programma's voor de PC. Ga het rijtje maar na. VIP Professional is een minstens even krachtige en zelfs snellere spreadsheet dan Lotus 1-2-3. Hetzelfde geldt voor DB-Man ten opzichte van dBase-III. En sinds kort is er Adimens, een relationele database die volledig GEM-ondersteund werkt en daarbij ook nog eens razendsnel is. Datzelfde pakket kostte voor Wang-computers, waarvoor het oorspronkelijk is ontwikkeld, vele duizenden guldens.”

- Maar voor de belangrijkste zakelijke toepassing, tekstverwerking, ben ik nog geen programma tegengekomen dat de mogelijkheden van de ST volledig benut.

De Graaf: „Daar komt dan binnenkort verandering in. Allereerst is nu een volledige Nederlandse versie van 1st Word Plus beschikbaar. Het meest opvallende daarvan is dat grafieken en tekeningen in de tekst kunnen worden opgenomen. Dan is er Write, de ST-versie van Microsoft Word, dat alom wordt beschouwd als een van de allerbeste tekstverwerkers. Dat programma is vooral geschikt voor mensen die heel omvangrijke documenten moeten schrijven; het heeft bijvoorbeeld zeer uitgebreide mogelijkheden voor fonts, pagina-indeling en voetnoten. En tenslotte wordt ook WordPerfect voor de ST uitgebracht.

Het is dus al lang niet meer zo dat er alleen door kleine softwarehuizen programma's voor de ST worden geschreven. Overigens, ook in Nederland wordt er opvallend goede software gemaakt. Bijvoorbeeld Projecta-4 en Factura-4, een compleet financieel administratiepakket voor het midden- en kleinbedrijf. Of Arkey, een programma waarmee architecten bouwontwerpen kunnen maken en zelfs complete bestekken kunnen doorrekenen. Er gebeurt echt heel veel.”

- Behalve de prijs, de kwaliteit van de hardware en software, is het vooral de service die een doorslaggevende rol speelt bij de aanschaf van een computersysteem. Daarover hoor ik veel klachten. Het lijkt me dat juist de service de zwakke schakel



is bij Atari.

De Graaf: „Service is in de eerste plaats een zaak van de dealer. In de adviesprijzen zit een redelijke marge verwerkt, waardoor de dealer ook aandacht kan besteden aan de after sales. Maar wie koopt bij een computerzaak die ver onder de adviesprijzen werkt, hoeft niet te rekenen op veel service van zo'n zaak. Dat kan ook niet, want daarvoor heeft de dealer z'n marge te klein gemaakt.”

- Maar hoort die dealer dan wel thuis in de Atari-organisatie? Als je de zakelijke markt wilt veroveren zul je er toch voor moeten zorgen dat je dealernet in orde is.

De Graaf: „Er worden dan ook regelmatig dealers van de lijst afgevoerd, die naar onze mening te weinig service verlenen of die klanten met een kluitje het riet in sturen. Als we daar herhaaldelijk klachten over krijgen, trekken we die na en als dan blijkt dat de klachten gegrond zijn, wordt het dealerschap ingetrokken. Gelukkig hebben we veel aanvragen van computerzaken, ook heel grote, die graag Atari-dealer willen worden. We hebben zelfs een dealerstop moeten invoeren.”

- U schuift de verantwoordelijkheid nu door naar de dealers, maar ook van hun

kant hoor ik klachten over Atari Benelux. Bijvoorbeeld over onderdelen die niet op voorraad zijn en over leveringsproblemen met de computers. Terwijl u eind vorig jaar uitdrukkelijk heeft gezegd dat de leveringsproblemen verleden tijd zouden zijn.

De Graaf: „Sommige dealers hebben inderdaad nog steeds leveringsproblemen. Dat heeft niet zozeer te maken met de produktiecapaciteit van de Atari-fabriek in Taiwan, want dat is een van de grootste ter wereld. Het heeft te maken met planning. De computers worden vanuit Taiwan verscheept naar Europa; daardoor zit er zo'n twee tot drie maanden tussen bestelling en aflevering. Vooral kleine dealers willen of durven niet zolang van tevoren te plannen, uit angst dat ze met een grote voorraad blijven zitten. Die zijn gewoon te voorzichtig; bij een actieve dealer worden de ST's verkocht als warme broodjes.”

- Als ik het goed begrijp kun je de goede dealer dus herkennen aan twee dingen: hij is niet de goedkoopste, en heeft voldoende ST's in z'n magazijn. Een aantal Atari-dealers heeft voorgesteld om de 'betere' dealers te onderscheiden van de rest door een organisatie van ster-dealers in het leven te roepen. Wat vindt u daarvan?


De Graaf: „Wij hebben er geen behoefte

aan om onderscheid te maken tussen onze dealers. De ST is een computer voor de massa, die moet je geen exclusieve status geven. En dus moet je ook geen net van exclusieve dealers opzetten.

Het karakter en de prijsstelling van de ST brengt met zich mee dat die zowel via hobby- als professionele dealers wordt verkocht. De eindgebruiker moet zelf uitmaken bij wie hij wil kopen: voor een bodemprijs bij een hobby-dealer, met het risico dat hij een slechte nazorg krijgt; of voor een paar honderd gulden extra bij een professionele dealer, waarbij hij dan ook moet kunnen rekenen op uitstekende service.

Toch zijn we daar de laatste tijd wat anders over gaan denken. Als het ontbreken van een professioneel dealernet tot problemen dreigt te leiden, overwegen we om de duurdere ST's en randapparatuur, zoals de laserprinter, alleen via professioneel ingestelde dealers te gaan verkopen. Maar zover is het, gelukkig, nog niet.”

■ Alfred Konijnenbelt



## SPECIAL DISK NR.5

### Sapristi, de labelspecialist

**ONTWERP LABELS** - vrije mogelijkheden om een eigen layout samen te stellen. Afmetingen, schriftsoorten, regelafstanden, centreren, enz. instelbaar. Ideaal voor een persoonlijk etiket.

**ETIKETTEN** met een vrije layout u maakt ze eenvoudig en snel met **SAPRISTI**

Alweer de vijfde schijf in onze speciale software-serie voor abonnees. En wat voor één: **SAPRISTI** is opgezet om het u heel wat makkelijker te maken. Eén van de meest praktische toepassingen van een computer is het maken van labels. **SAPRISTI** is daarin op drie manieren gespecialiseerd:

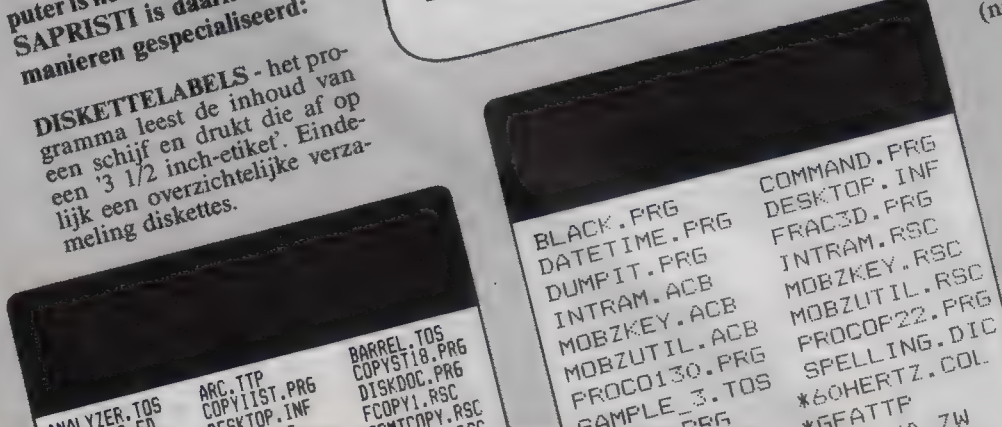
**ADRES LABELS** - het bekende etiket voor naam, adres, postcode en woonplaats, naar wens uitgebreid met telefoonnummer en een persoonlijk kenmerk. Inclusief database met uitgebreide sorteer- en zoekmogelijkheden.


**DISKETTELABELS** - het programma leest de inhoud van een schijf en drukt die af op een '3 1/2 inch-etiket'. Eindelijk een overzichtelijke verzameling diskettes.

**START**  
M. GIJZENBURG 14  
2907 HG CAPELLE a/d IJSSSEL

Verdere features: bijgeleverde RAM-disk voor extra snelle prestaties.

**SAPRISTI, DE LABEL-SPECIALIST** kost u als abonnee slechts f 39,- (niet-abonnees betalen f 49,-)







# Projecta-4 / Faktura-4 is boekhouden met plezier.

Debiteuren  
Krediteuren  
Kas, bank, giro  
Balans - V&W  
Voorraad  
Uren-registratie  
Projekt-administratie  
Fakturering

Projecta-4 / Faktura-4 is een van de plezierigste softwarepakketten op financieel gebied voor MS-DOS computers en de Atari-ST. Het heeft een ongekend groot gebruiksgemak en biedt standaard alles wat u van een serieus zakelijk pakket mag verwachten. Dat betekent dat u er voor nu en in de toekomst volledig mee uit de voeten kunt zonder eindeloos te moeten blijven investeren. Sterker nog: ons service-kontrakt geeft u het recht op gratis nieuwe versies als ons pakket straks nog uitgebreider wordt. En het geeft u tevens de zekerheid van een jaar lang uitstekende support! En dat alles voor een uiterst aangename prijs. Geen wonder dat er nu al honderden tevreden gebruikers zijn!

## UIT DE VAKPERS:

Het is natuurlijk bijzonder aardig om positieve reacties te krijgen op een produkt waarin je als maker gelooft. Zeker als de gebruikers - maar ook de vakmensen - onze filosofie achter het produkt blijken te herkennen.

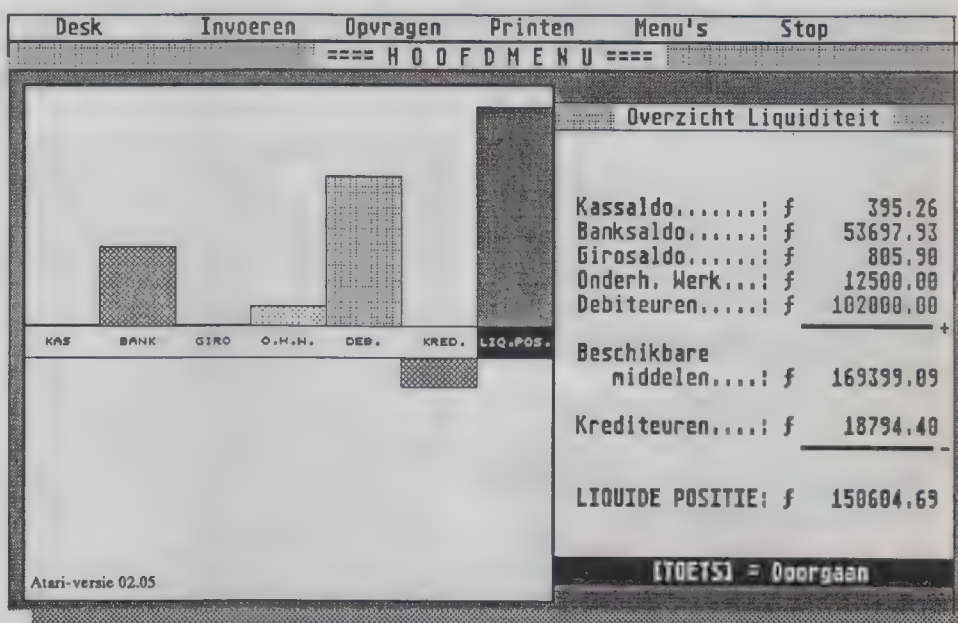
Onder het kopje 'Projecta-4 - Management-gericht' schrijft P.C.M., het toonaangevende computerblad:

*"De handleiding van Projecta-4 wint met mijlen voorsprong de prijs voor het mooiste uiterlijk en ook ruimschoots die voor de best geschreven tekst (...)"*  
P.C.M. mei 1987

Uit START (Atari-blad) mei 1987:

*"Het is een werkelijk een plezier om de dagelijkse boekingen in te voeren. (...) Fouten maken is nauwelijks mogelijk. (...)"*

Uit Atari-magazine, april 1987:  
*"Deze Rolls Royce onder de boekhoudprogramma's is voor iemand die meer dan een kas-, giro-, en bankboek wil een MUST en zijn geld dubbel en dwars waard."*



## NIEUWS

Projecta-4 heeft nu standaard het decimale rekeningschema met zescijferige vrije grootboeknummers.

Nieuw is de mogelijkheid om in Faktura de faktuur te verlengen met een acceptgirokaart, waarop alle gewenste gegevens automatisch worden ingevuld.

Voor administratiekantoren is nu de licentie-versie beschikbaar waarmee een onbepaald aantal boekhoudingen kan worden gevoerd.

Aan Faktura-4 is een extra optie toegevoegd van offertes en orderbevestigingen. Daarbij is het uiteraard mogelijk offertes direkt om te zetten in facturen, maar indien nodig kunnen ze ook worden gewijzigd en aangevuld en zelfs worden omgezet in een

regulier 'lopend' project waarop diverse andere leveringen kunnen worden geboekt.

In voorbereiding is een versie voor de gespecialiseerde boekhandel met een zeer uitgebreid ge-indexeerd artikelenbestand en zoekmogelijkheden op talloze velden.

Projecta-4 en Faktura-4 zijn nu ook verkrijgbaar in België in een versie die volledig is aangepast aan de Belgische wettelijke normen.

Eerst zien en dan geloven? Stap binnen bij een van de ca. 120 dealers in Nederland en België en vraag vrijblijvende demonstratie. Of bel TOPIC Software Amsterdam: 020-681400 voor een dealerlijst.

**TOPIC Software - Ruyschstraat 23 - 1091 BR Amsterdam**



# PROGRAMMA'S MET KARAKTER

Een truukje wat u beslist niet onthouden mag worden, is het veranderen van de karaktersets die de ST gebruikt. De karaktersets heten bij de ST de systeemfonts.

Er zijn er drie: het 6x6 systeemfont (dat in combinatie met de ikonen gebruikt wordt), het 8x8 systeemfont (normale tekst in lage resolutie) en het 8x16 systeemfont (normale tekst in hoge resolutie).

Als de systeemfonts worden veranderd, verandert bijna alles. Drop-down menu's, alertboxen, kortom alles wat tekst gebruikt zal uw eigen karakter-set tonen.

Dit geeft u bijvoorbeeld de mogelijkheid om meerdere fonts in een tekenprogramma op te nemen. Nadat u dit artikel grondig heeft bestudeerd, kunt u het tekenprogramma "KunST" (uit START 4) dus uitbreiden.

Of wat te denken van een spelletje? Niets leuker dan een eigen karakterset voor de hi-score tabel. Kortom: als u uw programma's iets speciaals wilt meegeven, is een eigen font een echte aanrader.

## GEM en TOS

Er zijn twee soorten programma's voor de ST: programma's die onder TOS draaien en programma's die van een GEM-omgeving (vensters, menu's) gebruik maken. In beide soorten programma's kan met zelf gedefinieerde karakters gewerkt worden.

Voor allebei de "toestanden" krijgt u in dit

artikel voldoende handvaten: er volgen maar liefst zes listings op de volgende bladzijden. De eerste listing werd geschreven door Karl Moens; listing twee tot en met zes zijn van de hand van Egon Lubbers.

## LISTING 1: FONT-ED.BAS

De eerste listing bevat een complete FONT-EDITOR. Met dit programma kunt u zelf karaktersets ontwerpen. Uiteraard kunt u ook bestaande fonts laden en wijzigen. De font-editor werkt alleen in combinatie met een monochrome monitor. De fonts zijn echter ook in de lagere resoluties te gebruiken. In aansluiting op dit artikel vindt u een complete gebruiksaanwijzing.

## LISTING 2: FONTMANI.BAS

Als we optimaal van fonts gebruik willen maken, hebben we de adressen nodig die het begin van de systeemfonts in het geheugen aangeven.

FONTMANI.BAS is een BASIC-programma dat een stukje machinetaal bevat om die "pointers" te vinden.

Deze listing vormt het hart voor onze volgende experimenten en moet zowel voor listing drie als voor listing vier geplaatst worden.

## LISTING 3: FONTDEMO.BAS

Dit programma demonstreert het gebruik van GEM- en TOS-fonts in GfA-BASIC op een monochroom monitor. U moet deze listing onder listing twee toevoegen en pas daarna "runnen".

Het programma rekent allereerst het 8x16 font om tot een "streepjes-font". Dat geldt zowel voor gebruik in TOS (de gewone opdracht PRINT) als GEM (de instructie TEXT). Na een druk op een toets wordt alles weer normaal. Weer op een toets drukken brengt een alertbox op het scherm. Er zijn nu twee dingen veranderd. Het TOS-font staat ingesteld op 8x8 karakters en het scherm scrollt met één puntje tegelijk (een extraatje). Binnen de alertbox is het 8x8 GEM-font actief. U kunt het met een muisklik op 6x6 zetten.

De keuze voor "exit" brengt een volgende alertbox. U kunt het programma afbreken met "stop". Hierdoor blijven zowel het 8x8-font als de één-punt-scroll actief. Als u kiest voor "cont", zet het programma de oude fonts weer terug en keert het netjes terug naar de editor.

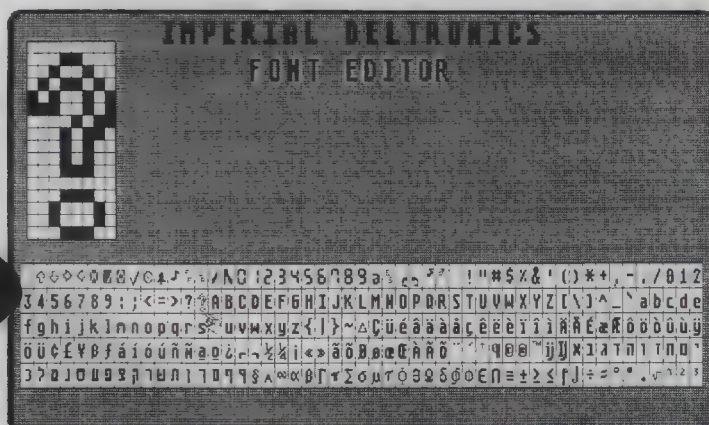
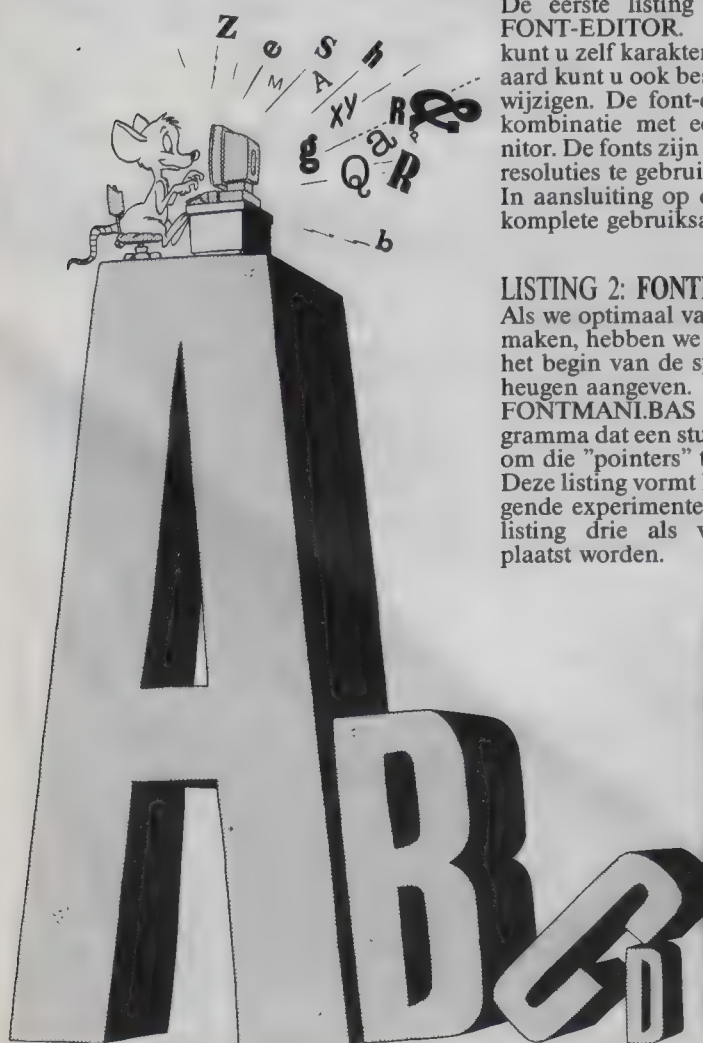
Als u een eigen karakterset in een GfA programma wilt gebruiken, is listing drie eveneens waardevol. Op de plaats waar het programma het streepjes-font ontwerpt, moet u het font inlezen. In de listing kunt u voor een willekeurig font FONT.FDA de juiste manier vinden. Dit stukje programma is echter met een apostrof buiten werking gesteld. Als u het wilt gebruiken, moet u de apostrof verwijderen en het streepjes-font uitschakelen.

## LISTING 4: DEGFDA.BAS

Bij DEGAS, waarschijnlijk het meest verspreide tekenpakket voor de ST, worden losse fonts geleverd. Het zou prettig zijn als we die in onze eigen programma's konden gebruiken. Een DEGAS-font bestaat uit 128 karakters van 16 bytes elk. De opbouw is anders dan bij de systeemfonts (zie verderop). Eerst wordt karakter 0 helemaal gedefinieerd, dan helemaal karakter 1, enz. De grootte van de complete karakterset is 128 karakters \* 16 bytes = 2048 bytes. Een utility om een DEGAS-font te converteren naar een direkt bruikbaar font is afgedrukt in listing vier. De fonts worden op schijf gezet met de extensie FDA.

Ook deze listing moet worden voorafgegaan door listing twee.

Nu we het toch over DEGAS hebben: de editor die daarbij wordt geleverd is ook prima geschikt om eigen karaktersets te ontwerpen. Converteer het resultaat dan (weer met listing vier) naar een FDA file. Hierna kunt u het font met listing drie in een GfA programma gebruiken of met listing zes in uw desktop installeren.





## LISTING 5: FONTDEMO.C

Natuurlijk zitten we voor het gebruik van fonts niet aan BASIC vast. Listing vijf demonstreert het gebruik van het 8x8 TOS-fonts in C. Compileer de source met Megamax-C.

## LISTING 6: SET-FONT.ASM

We eindigen met een programma in machinetaal. Geef de geassembleerde versie van deze source de naam SET-FONT.TTP. Als u het programma start, vraagt het om parameters. U voert de naam in van het font dat u wilt gebruiken (bijvoorbeeld GRIEKS.FDA). Uw desktop komt nu op met het nieuwe GEM-font (zie illustraties).

## WIJZERS VERDRAAIEN

Natuurlijk wilt u ook iets over de interne werking van de programma's weten. Laten we maar snel met de theorie beginnen. Een GEM-font bestaat uit een fonthead, fontdata, character- en een horizontaal offset table. Voor ons projekt zijn de fontheaders en de fontdata van de drie systeemfonts belangrijk.

Ten eerste moeten we er achter zien te komen op welke plaats de systeemfonts in het geheugen staan. Deze informatie is alleen te vinden met behulp van een stukje machinetaal (listing 2).

Dit is wat er gebeurt: de machinetaal voert de opcode \$A000 uit, waarna in A0 het startadres van de line A variabelen staat. In A1 vinden we een vektor die naar drie pointers op de "systeemfontheaders" wijst. Laten we deze fontheaders eens nader bekijken.

Een fonthead beschrijft hoe een font er uitziet (o.a. grootte, breedte en vorm van de karakters in dat font). Een belangrijk onderdeel is de pointer die aangeeft op welke plaats de fontdata beginnen. De

B:\			C:\		
73476 bytes used in 21 items			642970 bytes used in 59 items		
DEGFOR	BAS	2294	GROENBO	DOC	5141
FONTCON	BAS	5038	VRAAG	FNT	4096
FONTDEMO	BAS	3036	SETUP	IC	1492
SPRITEZ	BAS	2376	DESKTOP	INF	692
CLASSIC	FDA	4096	ACCLDAD	KOL	6834
CURSIVE	FDA	4096	FLOP	KOL	4796
KINDER	FDA	4096	MOENS	KOL	14108
LUGUBER	FDA	4096			
MASH	FDA	4096			
SHADOW	FDA	4096			
STENCIL	FDA	4096			
SYSTEM	FDA	4096			
TEIN	FDA	4096			
DESKTOP	INF	478			
FONTCON	PRG	11772			
FONTDEMO	TOS	5893			
SET_FONT	TTP	4243			

eerste byte van karakter 2, enz... Na van alle 256 karakters de eerste byte te hebben gedefinieerd volgt de tweede byte van karakter 0, dan weer de tweede byte van karakter 1, enz. (zie figuur 1). Een 8x16 font is dus 4096 bytes groot (256 karakters van elk 16 bytes), een 8x8 font 2048 bytes. Het 6x6 font is 256\*6\*6 bits groot, dus 256\*6\*6/8 = 1152 bytes. Houd er in dit laatste geval rekening mee dat alle bits gebruikt worden. (Dus 1 byte bevat data voor 8/36 = 2/9 karakter.)

Laten we even terugkeren naar de drie pointers die naar de systeemfonts wijzen. Pointer 1 wijst naar de header van het 6x6 systeemfont, pointer 2 naar de header van het 8x8 systeemfont en pointer 3 op de header van het 8x16 systeemfont.

Helaas moet ons plan in de prullenbak: de gebruiker kan deze pointers niet op zijn eigen fontheaders laten wijzen, omdat ze in ROM staan. Datzelfde geldt voor de pointers naar de fontdata. Alle pointers die niet gewijzigd kunnen worden, staan met geheel zwarte lijnen in figuur 2 getekend. De pointers die de gebruiker wel kan veranderen zijn met gestippelde lijnen aangegeven.

## DUBBELE INFORMATIE

Maar er was nog een pointer; deze wees op de "line A variabelen". Deze variabelen bevatten informatie voor de teken- en tekstfuncties.

Na enige studie bleek dat GEM en TOS niet direkt naar de fonts in ROM verwijzen, maar dat ze van de systeemfontheaders een werkkopie in RAM hebben gemaakt. Om de pointers naar deze in RAM staande fontheaders te vinden, komt die tweede variabele van pas. De pointers naar deze fontheaders zijn te vinden op:

- \* A0 - 460: pointer op de 8x16 fonthead in RAM
- \* A0 - 456: pointer op de 8x8 fonthead in RAM
- \* A0 - 452: pointer op de 6x6 fonthead in RAM

(En A0 is hierbij het adres van de "line A variabelen")

Het heeft geen zin om deze pointers te veranderen: ze zijn alleen ter informatie en worden verder niet gebruikt. De 'echte' fontheadpointers staan in ROM. Weer een aanpak die mislukt.

## DIK TEVREDEN

Dan is de enige oplossing voor ons probleem om over de bestaande fonthead in RAM een nieuwe fonthead te zetten. Meestal is het niet nodig om een font (inclusief header, character- en horizontaal offset table en fontdata) te schrijven. Als alleen de fontdata veranderd worden, zijn we al dik tevreden. Het gevolg van het veranderen van alleen de fontdata is dat het font z'n oorspronkelijke afmetingen behoudt, maar een ander uiterlijk krijgt.

Zo is het bijvoorbeeld mogelijk om Griekse of horror-letters te maken. Om het 8x16 systeemfont door de zelfgedefinieerde fontdata te vervangen, is de volgende GfA-instructie nodig:

Lpoke Lpeek(Lnavar% - 460)+76, Varptr(Eigenfont\$)

Waarbij Lnavar de variabele is die via de routine van listing twee kan worden verkregen en waarbij Eigenfont\$ een string van 4096 bytes groot is waarin de fontdata staan (zie listing drie).

Natuurlijk kan dit ook in de taal C:

\* (long \*) (76 + \*(long \*) (lnavar-460)) = (long) &Eigenfont;

We krijgen het systeemfont later weer terug door:

\* (long \*) (76 + \*(long \*) (lnavar-460)) = \* (long \*) (h8x16+76);

Hier is "Eigenfont" bijvoorbeeld gedeclareerd als

char Eigenfont[4096];

Het GEM font wordt in C bij de functie 'v-gtext()' en in GfA bij de functie 'text' gebruikt. Alle bekende effecten (bold, cursief, vergroten) blijven na het veranderen van de fontdata mogelijk. Rest nu nog een ander soort font: het TOS-font.

## KLEINE LETTERS

Programma's met de extensie TTP of TOS maken geen gebruik van GEM. Ze zijn daardoor wat sneller in hun tekstbehandeling. TOS maakt dan ook geen gebruik van fontheaders, maar van een viertal andere variabelen.

- \* A0 - 22: fontdata pointer
- \* A0 - 40: aantal bytes opschuiven voor een nieuwe regel
- \* A0 - 42: aantal regels op het beeldscherm -1 (normaal 24)
- \* A0 - 46: hoogte van de karakters

FIGUUR 1

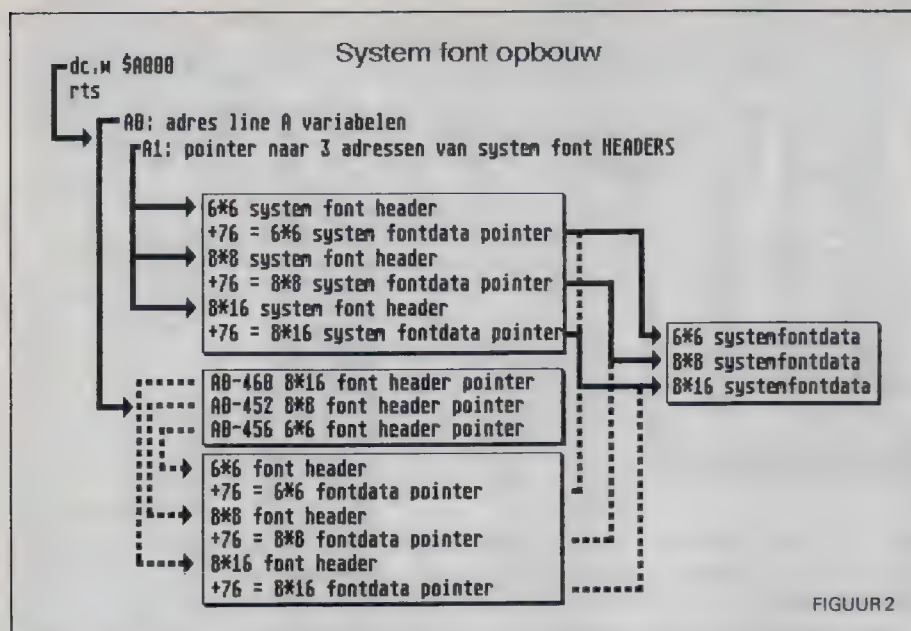


Opbouw fontdata

fontdata beschrijven hoe de karakters van het font er uit komen te zien. Deze pointer staat 76 bytes na het begin van de fonthead. Als we kans zien de fontdata pointer te veranderen, kunnen we hem naar elke plek in het geheugen laten wijzen. Als we op een bepaalde plaats onze eigen fontdata neerzetten, hebben we bereikt wat we willen. (De grootte en breedte van het font hoeven we niet te veranderen: deze liggen immers al vast (6x6 onder ikonen, 8x8 in kleur en 8x16 in monochroom).

Natuurlijk moeten we vervolgens de opbouw van de fontdata bestuderen. Deze is een beetje ingewikkeld: de eerste byte is de eerste byte van karakter 0. Vervolgens volgt de eerste byte van karakter 1, dan de





Het TOS-font kan in GfA via 'print' en in C via 'printf()' worden aangeroepen. Het is niet mogelijk om het TOS-font bijvoorbeeld kursief te laten verschijnen. Dit kan alleen met een GEM-font. In monochroom zal A0 - 22 een pointer bevatten naar het 8x16 systeemfont, in kleur naar het 8x8 font.

Er bestaat een leuk truukje om in monochroom niet het 8x16 font te gebruiken, maar het 8x8 font. Buiten het instellen op de 8x8 fontdata is het noodzakelijk om de hoogte van de karakters aan te passen (8) en het aantal regels en op te schuiven bytes te veranderen (bijvoorbeeld in respectievelijk 49 en 640). Deze methode wordt onder andere in de GfA-BASIC interpreter toegepast bij 'text 16' en 'text 8'. In de listings drie en vijf ziet u andere voorbeelden.

Het is niet mogelijk om het 6x6 systeemfont als TOS-font te gebruiken: de breedte van de karakters moet altijd 8 bits zijn.

### Egon Lubbers

### LISTING 1

```

*****
*                FONT EDITOR                *
*                door Karl Moens             *
*****
If Xbios(4)<>2 Then
  Setcolor 3,7,0,0
  Setcolor 0,0,7,0
  Alert 3,"The FONT-EDITOR needs! hi =>
gh resolution.",1,"Sorry",Dummy
  Setcolor 3,0,0,0
  Setcolor 0,7,7,7
  End
Endif
On Break Gosub Quit
Gosub Find_system_font
Dim FontX(1024),Men_tree$(40),Help$(2 =>
4),Edit_field$(9,17)
Arrayfill FontX(),0
Arrayfill Edit_field(),0
For T=0 To 21
  Read Help$(T)
Next T
Data ""
Data "Load a font from diskito be edi =>
ted."
Data "Save a font on diskito be used =>
later."
Data "This BASIC-program will allowly =>
ou to use the new fonts!in your own p =>
rograms."
Data ""
Data "Move a character into:the edit =>
field.!!Just click on the character!yo =>
u want to edit."
Data "Move a character from:the edit =>
field into the!font table."
Data "Delete the character in the!edi =>
t field."
Data "Black becomes white.!!White beco =>
mes black."
Data "Shift the character in the!edit =>
field 1 pixel.!!Use the arrow keys.!!C =>
lick to end."
For T=0 To 40
  Read Men_tree$(T)
  Exit If Men_tree$(T)="***"
Next T
Gosub Set_system_font
Menu Men_tree$(1)
Gosub Find_system_font
Read Off
While Off<>-1
  Menu Off,2
  Read Off
Wend
Data DESK, Font Editor ,----- =>
-----,,-,-,-,-,"
Data FILE, Load Font , Save Font , =>
Save Program , Quit ,""
Data EDIT, Load Char , Save Char , =>
Delete Char , Inverse Char , Shift =>
Char ,""
Data HELP, More Help ,""
Data 12,18,19,20,21,-1
Deffill 1,2,4
Prbox 5,25,635,395
Define 1,3

```

```

Rbox 5,25,635,395
Define 1,1
Xbox=20
Ybox=80
Unit=10
Deffill 1,0
Pbox Xbox,Ybox,Xbox*8*Unit,Ybox+16*Unit
Graphmode 2
Defext 1,1,0,20
Text 140,50,340,"IMPERIAL DELTRONICS"
Defext 0,16,0,20
Text 220,80,180,"FONT EDITOR"
Defext 1,0,0,20
Text 219,80,180,"FONT EDITOR"
Graphmode 1
For X=1 To 7
    Line Xbox*Unit*X,Ybox,Xbox*Unit*X,Y =>
box*Unit+16
Next X
For Y=1 To 16
    Line Xbox,Ybox+Unit*Y,Xbox+Unit*8,Y =>
box*Unit+Y
Next Y
Pbox 14,260,626,360
For X=1 To 50
    Line 14+X*12,260,14+X*12,360
Next X
For Y=1 To 4
    Line 14,260+Y*20,626,260+Y*20
Next Y
Font_adres=Varptr(FontZ(0))
Gosub Set_user_font
Gosub Show_font
Gosub Set_system_font
Let Potload$=Mki$(0)+Mki$(15)+Mki$(1)
Let Potload$=Potload$+Mki$(0)+Mki$(1)
For IX=1 To 16
    Read Hinten
    Let Potload$=Potload$+Mki$(Hinten)
Next IX
For IX=1 To 16
    Read Vorn
    Let Potload$=Potload$+Mki$(Vorn)
Next IX
Data 30,63,127,255,511,1022,2044,4088 =>
,8176,16352,32704,65408,65280,65024,6 =>
4512,63488
Data 0,28,38,74,146,292,584,1168,2336 =>
,4672,15488,17664,17920,25600,30720,0
On Menu Ibox 1,Xbox,Ybox,Unit*8,Unit* =>
16 Gosub Tekan_start
On Menu Ibox 2,14,260,612,100 Gosub K =>
ies_start
On Menu Gosub Keuze
On Menu Message Gosub Mouse_click
Do
    On Menu
    If Mousek
        Gosub Mouse_click
    Endif
Loop
Procedure Tekan_start
    Defmouse Potload$
    Let In_box_flag=1
    On Menu Obox 1,Xbox,Ybox,Unit*8,Uni =>
t*16 Gosub Tekan_end
Return
Procedure Tekan_end

```

```

Defmouse 0
  Let In_box_flag=0
  On Menu Ibox 1,Xbox,Ybox,Unit*8,Uni =>
t:16 Gosub Tekan_start
Return
Procedure Kies_start
  Defmouse 3
  On Menu Obox 2,14,260,612,100 Gosub =>
Kies_end
  Let In_box_flag=2
  If Char_edit_flag=0
    Alert 2,"Do you want to move! =
a character! TO or FROM: the Edit =>
Field?",1,"To:From",Char_edit_flag
    Font_edited_flag=1
    Defmouse 3
    Menu 16+Char_edit_flag,1
    Menu 19-Char_edit_flag,0
  Endif
Return
Procedure Kies_end
  Defmouse 0
  Let In_box_flag=0
  On Menu Ibox 2,14,260,612,100 Gosub =>
Kies_start
Return
Procedure Mouse_click
  If In_box_flag
    Klik=Mousek
    If In_box_flag=1
      Pos_x=Int((Mousex-Xbox)/Unit)+1
      Pos_y=Int((Mousey-Ybox)/Unit)+1
      If Pos_x>0 And Pos_x<9 And Pos_y =>
y>0 And Pos_y<17
        If Klik=1
          Let Edit_field%(Pos_x,Pos_y)=1
          Deffill 1
          Pbox Xbox+(Pos_x-1)*Unit+1, =>
Ybox+(Pos_y-1)*Unit+1,Xbox+Pos_x*Unit =>
-1,Ybox+Pos_y*Unit-1
          Endif
          If Klik=2
            Let Edit_field%(Pos_x,Pos_y)=0
            Deffill 0
            Pbox Xbox+(Pos_x-1)*Unit+1, =>
Ybox+(Pos_y-1)*Unit+1,Xbox+Pos_x*Unit =>
-1,Ybox+Pos_y*Unit-1
            Endif
          Endif
        Else
          Code_char=Int(((Mousex-14)/12)+5) =>
1*Int(((Mousey-260)/20))
          If Char_edit_flag=1
            Gosub Set_edit_field(Code_char)
            Gosub Show_char
          Else
            Gosub Show_font(Code_char)
          Endif
        Endif
      Endif
    Endif
Return
Procedure Keuze
  Keuze=Menu(0)
  If More_help_flag
    On Keuze Gosub Du,Du,Du,Du,Du,Du,Du, =>
Du,Du,Du,Du,Du,Help,Help,Help,Du,Du,Du,H =>
elp,Help,Help,Help,Help,Du,Du,Du
  Endif

```



```

On Keuze Gosub Font_editor,Du,Du,Du =>
,Du,Du,Du,Du,Du,Du,Load_font,Save_font =>
t,Save_program,Quit,Du,Du,Load_char,S =>
ave_char,Delete_char,Inverse_char,Mov =>
e_char,Du,Du,More_help
Return
Procedure Du
Return
Procedure Help
Alert 0,Help$(Keuze),1,"OK!Cancel", =>
Antwoord
If Antwoord=2
Keuze=2
Endif
Return
Procedure More_help
Alert 2,"Do you want Help-messages! =>
to appear automatically?",1,"Yes!No", =>
Antwoord
If Antwoord=1
More_help_flag=1
Menu 23,1
Else
Menu 23,0
More_help_flag=0
Endif
Return
Procedure Font_editor
Alert 0," IMPERIAL DELTRONICS! =>
Presents! a F.I.R.M.A. program!F 0 =>
N T E D I T O R",1,"= 1986",Antwoord
Return
Procedure Load_font
Antwoord=1
If Font_flag
Alert 1,"Loading a new font will! =>
replace the existing one.",1,"OK!Abor =>
t",Antwoord
Endif
If Antwoord=1
Fileselect "\*.fnt","SYSTEM.FNT", =>
Font_naam$
If Font_naam$=""
Goto Proc_load_font_einde
Endif
If Not (Exist(Font_naam$))
If Right$(Font_naam$,10)<>"SYST =>
EM.FNT"
Alert 2,"This font does not e =>
xist!Do you want to create it?",1,"Y =>
es!No",Antwoord
If Antwoord=2
Goto Proc_load_font_einde
Endif
Arrayfill Font$(0),0
Else
Font_adres=Varptr(Font$(0))
For T=0 To 4095 Step 4
Lpoke Font_adres+T,Lpeek(Sy =>
s_font+T)
Next T
Endif
Else
Font_adres=Varptr(Font$(0))
Bload Font_naam$,Font_adres
Endif
Font_flag=1
Font_saved_flag=1
Font_edited_flag=0
Char_ascii=0
Char_edit_flag=0
Gosub Set_user_font
Gosub Show_font
Arrayfill Edit_field$(0),0
Gosub Show_char
Gosub Set_system_font
Menu 12,3
Menu 18,3
Menu 19,3
Menu 20,3
Menu 21,3
Menu 17,0
Menu 18,0
Endif
Proc_load_font_einde:
Return
Procedure Show_all_char
For T=0 To 254
Gosub Set_edit_field(T)
Gosub Show_char
Next T
Return
Procedure Show_font
Deftext 1,0,0,13
Graphmode 1
For Y=0 To 4
For X=0 To 50
Text 17+X*12,276+Y*20,Chr$(Y*51+X)
Next X
Next Y

```

```

Return
Procedure Find_system_font
Let Sys_font=Lpeek(&H607E)
Return
Procedure Set_user_font
Lpoke &H607E,Font_adres
Return
Procedure Set_system_font
Lpoke &H607E,Sys_font
Return
Procedure Show_char
Deffill 1,2,8
For YX=1 To 16
For XZ=1 To 8
If Edit_field$(XZ,YX)
Deffill 1
Pbox Xbox+(XZ-1)*Unit+1,Ybox+ =>
(YZ-1)*Unit+1,Xbox+XZ*Unit-1,Ybox+YZ* =>
Unit-1
Else
Deffill 0
Pbox Xbox+(XZ-1)*Unit+1,Ybox+ =>
(YZ-1)*Unit+1,Xbox+XZ*Unit-1,Ybox+YZ* =>
Unit-1
Endif
Next XZ
Next YX
Return
Procedure Set_edit_field(Code)
Font_adres=Varptr(Font$(0))
Start_adres=Font_adres+Code
For TX=0 To 15
Muster$=Right$("00000000"+Bin$(Pe =>
ek(Start_adres+TX*256)),8)
For TIX=1 To 8
Let Edit_field$(TIX,TX+1)=Val(M =>
id$(Muster$,TIX,1))
Next TIX
Next TX
Return
Procedure Change_font(Code)
X=Int((MouseX-14)/12)
Y=Int((MouseY-260)/20)
Font_adres=Varptr(Font$(0))
Start_adres=Font_adres+Code
For TX=0 To 15
T$=""
For TIX=1 To 8
T$=T$+Str$(Edit_field$(TIX,TX+1))
Next TIX
Poke Start_adres+TX*256,Val(T$)
Next TX
Gosub Set_user_font
Deftext 1,0,0,13
Graphmode 1
Text 17+X*12,276+Y*20,Chr$(Code)
Gosub Set_system_font
Return
Procedure Save_font
Antwoord=1
If Font_edited_flag=0 And Font_save =>
d_flag=1
Alert 1,"This font has not been e =>
dited!since it was last saved.",2,"Co =>
ntinue!Abort",Antwoord
Endif
If Antwoord=1
Fileselect "\*.fnt","USER.FNT",Fa =>
nt_save_naam$
If Font_save_naam$=""
If Exist(Font_save_naam$)
Alert 1,"A font with this nam =>
e already exists!It will be overwrit =>
ten.",2,"OK!Abort",Antwoord
If Antwoord=2
Goto Proc_save_font_einde
Endif
Endif
Font_adres=Varptr(Font$(0))
Bsave Font_save_naam$,Font_adre =>
s,4096
Font_edited_flag=0
Font_saved_flag=1
Font_naam$=Font_save_naam$
Endif
Proc_save_font_einde:
Return
Procedure Load_char
Char_edit_flag=1
Font_edited_flag=1
Menu 17,1
Menu 18,0
Return
Procedure Save_char
Char_edit_flag=2
Font_edited_flag=1
Menu 17,0
Menu 18,1
Return

```

```

Procedure Quit
If Font_saved_flag=0 And Font_edite =>
d_flag=1
Alert 1,"This font has not!yet be =>
en saved.",1,"Save!Quit!Abort",Antwoord
If Antwoord=1
Gosub Save_font
Gosub Quit
Endif
If Antwoord=2
Antwoord=1
Gosub Einde
Endif
Else
Antwoord=2
Gosub Einde
Endif
Return
Procedure Einde
If Antwoord<>1
Alert 2,"Are you sure!you want to =>
quit?",2,"Yes!No",Antwoord
Endif
If Antwoord=1
Gosub Set_system_font
Deffill 1,2,8
Graphmode 1
For T=1 To 32
Pbox 320-T*10,200-T*6.5,320+T*1 =>
0,200+T*6.5
Next T
End
Endif
Return
Procedure Delete_char
Alert 2,"Are you sure you want!to d =>
elete this character?",2,"Yes!No",Ant =>
woord
If Antwoord=1
Arrayfill Edit_field$(0),0
Gosub Show_char
Endif
Return
Procedure Inverse_char
For TX=1 To 16
For TIX=1 To 8
If Edit_field$(TIX,TX)
Let Edit_field$(TIX,TX)=0
Else
Let Edit_field$(TIX,TX)=1
Endif
Next TIX
Next TX
Gosub Show_char
Return
Procedure Move_char
Null$=Chr$(0)
Proc_move_char_start:
Repeat
A$=Inkeys
Until A$<>" " Or MouseX
If MouseX Or A=13
Goto Proc_move_char_einde
Else
If A$=Null$+"H"
For TyZ=0 To 15
For TxZ=1 To 8
Let Edit_field$(TxZ,TyZ)=Ed =>
it_field$(TxZ,TyZ+1)
Next TxZ
Next TyZ
For TxZ=1 To 8
Let Edit_field$(TxZ,16)=Edit_ =>
field$(TxZ,0)
Next TxZ
Endif
If A$=Null$+"P"
For TyZ=17 To 2 Step -1
For TxZ=1 To 8
Let Edit_field$(TxZ,TyZ)=Ed =>
it_field$(TxZ,TyZ-1)
Next TxZ
Next TyZ
For TxZ=1 To 8
Let Edit_field$(TxZ,1)=Edit_f =>
ield$(TxZ,17)
Next TxZ
Endif
If A$=Null$+"M"
For TyZ=1 To 16
For TxZ=9 To 1 Step -1
Let Edit_field$(TxZ,TyZ)=Ed =>
it_field$(TxZ-1,TyZ)
Next TxZ
Next TyZ
For TyZ=1 To 16
Let Edit_field$(1,TyZ)=Edit_f =>
ield$(9,TyZ)
Next TyZ
Endif

```



# GEBRUIKSAANWIJZING FONT~EDITOR

Dit programma stelt u in staat om de karakterset van de ST te veranderen. Het programma is alleen geschikt voor gebruik met een monochrome monitor. De fonts die het produceert, zijn echter geschikt voor alle resoluties.

De font-editor maakt gebruik van een standaard menu-balk. Hierin treft u de volgende mogelijkheden aan:

## DESK MENU

**FONT EDITOR:** Er verschijnt een alertbox met een copyright mededeling.

## FILE MENU

**LOAD FONT:** Via een zogenaamde 'file selector' kunt u een font van een diskette in het geheugen laden. Wanneer u nog geen font bezit, kunt u altijd de standaard karakterset van de ST laden. Deze haalt het programma namelijk uit ROM. De file selector biedt uit zichzelf de keuze voor SYSTEM.FNT al aan. U hoeft in dat geval dus alleen op "OK" te klikken.

**SAVE FONT:** Door middel van deze functie kunt u een zelfgemaakte karakterset op een schijf vastleggen. Als u '.FNT' aan de bestandsnaam toevoegt, zal de functie LOAD FONT uw karakterset in het vervolg automatisch in de file selector plaatsen.

**SAVE PROGRAM:** Deze functie schrijft een GfA-BASIC utility op uw schijf. Hierdoor kunt u gemakkelijk een zelfgemaakte karakterset in een eigen programma opnemen.

Zorg ervoor dat de routine in uw program-

ma als eerste wordt uitgevoerd. U moet zelf aangeven hoeveel fonts u wilt gebruiken en wat de namen van de fonts zijn.

Dus:  
[aantal fonts], [naam font 1], [naam font 2] etc.

In uw eigen deel van het programma activeert u een bepaalde karakterset met de opdracht

Gosub Set-user-font(font nummer)  
Alle karakters die via het BASIC-woord 'Text' naar het scherm gestuurd worden, zullen hierna in uw eigen font worden weergegeven. U moet de hoogte van de karakters tenminste op dertien zetten.

**QUIT:** U keert terug naar de desktop.

## EDIT

**LOAD CHAR:** Klik op het karakter uit de tabel dat u wilt veranderen. Hierdoor komt er een kopie van het karakter in de editor te staan.

Ontwerpen doet u met de muis. De linkerknop maakt een pixel zwart; de rechter produceert wit.

**SAVE CHAR:** Kopieer het karakter uit de editor naar de tabel door op de gewenste bestemming te klikken.

**INVERSE CHAR:** Alle zwarte pixels van het karakter in de editor worden wit en vice versa.

**DELETE CHAR:** Wist de editor.

**MOVE CHAR:** Verschuift het karakter in de editor een pixel. De cursortoetsen bepalen de richting van de verschuiving. Door "enter", "return" of een muisklik wordt deze functie beëindigd.

## HELP

**MORE HELP:** Als deze functie actief is, toont het programma bij gebruik van andere opties voortaan alertboxen met korte instructies.

```
If A$=Null$+"K"
For TyZ=1 To 16
For TxZ=0 To 8
Let Edit_field%(TxZ,TyZ)=Ed =>
it_field%(TxZ+1,TyZ)
Next TxZ
Next TyZ
For TyZ=1 To 16
Let Edit_field%(8,TyZ)=Edit_f =>
ieldZ(0,TyZ)
Next TyZ
Endif
Endif
Gosub Show_char
Goto Proc_move_char_start
Proc_move_char_einde:
Return
Procedure Save_program
Fileselect "\*.lst","USER.LST",Prog =>
_save_naam$
If Prog_save_naam$<>""
If Exist(Prog_save_naam$)
Alert 1,"A program with this na =>
me!already exists.!It will be overwri =>
tten.",2,"OK!Abort",Antwoord
If Antwoord=2
Goto Proc_save_program_einde
Endif
Endif
Open "O",#1,Prog_save_naam$
Print #1;" "
Print #1;" IMPERIAL DELTRONICS F =>
ONT-EDITOR APPLICATION PROGRAM "
Print #1;" "
Print #1;" INITIALISATION"
Print #1;"Let Sys_font=Lpeek(&H60 =>
7E)"
Print #1;"Restore Font_data_section"
Print #1;"Read Number_of_fonts"
Print #1;"Dim FontZ(1024,Number_o =>
f_fonts-1)"
Print #1;"Dim Font_adres(Number_o =>
f_fonts-1)"
Print #1;" "
Print #1;" HERE FOLLOWS THE FONT =>
DATA SECTION WITH THE FOLLOWING FORMAT"
Print #1;" FIRST ENTRY : NUMBER =>
OF FONTS"
Print #1;" SECOND ENTRY : FILENA =>
ME OF FIRST FONT"
Print #1;" THIRD ENTRY : FILENA =>
ME OF SECOND FONT"
Print #1;" ETC. : ETC."
Print #1;"Font_data_section:"
Print #1;"DATA"
Print #1;" "
Print #1;" LOAD YOUR FONTS FROM =>
DISK"
Print #1;"FOR I=0 TO NUMBER_OF_FD =>
NTS-1"
Print #1;"Font_adres(I)=Varptr(Fo =>
ntZ(0,I))"
Print #1;"Read Font_name$"
Print #1;"Bload Font_name$,Font_a =>
dres(I)"
Print #1;"Next I"
Print #1;" "
Print #1;" CONTINUE WITH YOUR PR =>
OGRAM"
Print #1;"Goto Your_program_conti =>
nues_here"
Print #1;" "
Print #1;" HERE FOLLOW SOME USEF =>
UL PROCEDURES"
Print #1;" "
Print #1;" CALL THIS PROCEDURE B =>
EFORE USING YOUR OWN FONT"
Print #1;" ARGUMENT = FONT NUMBER"
Print #1;"Procedure Set_user_font =>
(Font_number)"
Print #1;"Lpoke &H607E,Font_adres =>
(Font_number-1)"
Print #1;"Return"
Print #1;" "
Print #1;" CALL THIS PROCEDURE W =>
HEN YOU WANT TO USE THE SYSTEM FONT"
Print #1;" AND IN ANY CASE BEFOR =>
E QUITTING YOUR PROGRAM"
Print #1;"Procedure Set_system_font"
Print #1;"Lpoke &H607E,Sys_font"
Print #1;"Return"
Print #1;" "
Print #1;" YOUR PROGRAM CONTINUE =>
S HERE"
Print #1;"Your_program_continues_ =>
here:"
Close #1
Endif
Proc_save_program_einde:
Return
```

Karl Moens

B:\				C:\			
73476 bytes used in 21 items				642970 bytes used in 59 items			
DECFON	BAS	2294		CRDENE80	DOC	5141	
FONTCON	BAS	5038		UTRAC	FNT	4096	
FONTDEMO	BAS	3036		SETUP	IC	1492	
SPTITE2	BAS	2376		DESKTOP	INF	692	
CLASSIC	FON	4096		ACCLORD	KOL	6834	
CURSIVE	FON	4096		FLOP	KOL	4796	
KINQER	FON	4096		MOENS	KOL	14108	
LUCUBER	FON	4096		ACCLORD	LST	6679	
MASH	FON	4096		FLOP	LST	4701	
SHADOW	FON	4096		MOENS	LST	13578	
STENCIL	FON	4096		CURSUS52	PC3	6054	
SYSTEM	FON	4096		CURSUS50	PI3	32034	
THIN	FON	4096		CURSUS51	PI3	32034	
DESKTOP	INF	478		FIGUW1	PI3	32066	
FONTCON	PRG	11772		FIGUW2	PI3	32066	
FONTDEMO	TOS	5893		BTW12	PRG	25509	
SET_FONT	TTP	4243		CURSUS5	PRG	455	



## LISTING 2

```
FONT MANIPULATING UTILITY                      Egon Lubbers, '87

Lnavar-22 : TOS- fontdata pointer
Lnavar-40 : (16 bits) TOS- aantal bytes opschuiven voor nieuwe regel
Lnavar-42 : (16 bits) TOS- aantal regels op het scherm - 1
Lnavar-46 : (16 bits) TOS- hoogte van de letters
Lnavar-460: GEM- 8*16 fonthead pointer
Lnavar-456: GEM- 6*6 fonthead pointer
Lnavar-452: GEM- 8*8 fonthead pointer
Lpeek(Lnavar-460)+76: GEM- 8*16 fontdata pointer (veranderbaar)
Lpeek(Lnavar-456)+76: GEM- 6*6 fontdata pointer ( " )
Lpeek(Lnavar-452)+76: GEM- 8*8 fontdata pointer ( " )
H6x6      : 6*6 systemfont header (in ROM)
H8x8      : 8*8 systemfont header (in ROM)
H8x16     : 8*16 systemfont header (in ROM)
Lpeek(H6x6+76) : 6*6 systemfont fontdata pointer (in ROM)
Lpeek(H8x8+76) : 8*8 systemfont fontdata pointer (in ROM)
Lpeek(H8x16+76): 8*16 systemfont fontdata pointer (in ROM)

Fontptr$=Mki$(&HA000)+Mki$(&H2009)+Mki$(&H4E75) ! 2 stukjes machinetaal
Lnavar$=Mki$(&HA000)+Mki$(&H4E75)             ! uitvoeren
Fontptr%=Varptr(Fontptr$)
Fontptr%=C:Fontptr$( )
Lnavar%=Varptr(Lnavar$)
Lnavar%=C:Lnavar$( )

H6x6=Lpeek(Fontptr%)
H8x8=Lpeek(Fontptr%+4)
H8x16=Lpeek(Fontptr%+8)
```

## LISTING 3

```
ZET FONT MANIPULATING UTILITY HIERVOOR:
Start monochroom font voorbeeld
Zet GEM font op een adres met eigen fontdata

Defext ,4                      ! GEM text in italics
Ffont$=Space$(4096)           ! ruimte voor font
Ffont%=Varptr(Ffont$)
Bmove Lpeek(H8x16+76),Ffont%,4096 ! kopieer 8*16 systemfont naar ons fontje
For X%=0 To 15 Step 2
    Poke Ffont%+X%*256+Y%,0      ! streepjes font maken
    Next Y%
Next X%
' Bload "FONT.FDA",Ffont%       ! methode om FDA-font te laden
Lpoke Lpeek(Lnavar%-460)+76,Ffont% ! verander 8*16 GEM fontdata pointer
Text 100,100,"ABC..." (GEM-font is veranderd)
Lpoke Lnavar%-22,Ffont%         ! verander TOS fontdata pointer
Print "ABC..." (TOS-font is veranderd)
Void Inp(2)
' Zet alles weer normaal
Cls
Lpoke Lpeek(Lnavar%-460)+76,Lpeek(H8x16+76)
Lpoke Lnavar%-22,Lpeek(H8x16+76)
Print "ABC..." (TOS-font weer normaal)
Text 100,100,"ABC..." (GEM-font weer normaal)
Void Inp(2)
' Verander TOS font in 8x8 font
Cls
Lpoke Lnavar%-22,Lpeek(H8x8+76) ! zet TOS fontdata pointer op 8*8 fontdata
Dpoke Lnavar%-46,8              ! 8 bits hoog
Dpoke Lnavar%-40,80             ! 80 bytes = 1 bit omlaag opschuiven
Dpoke Lnavar%-42,150            ! 150 regels van 1 bit
For X%=0 To 140
    Print "Soft scrolling 8x8 font"
Next X%
' Verander GEM font header
Gemheader$=Space$(88)
Bmove Lpeek(Lnavar%-460),Varptr(Gemheader$),88 ! kopieer header
A=2
Repeat
    If A=2
        Bmove H8x8+2,Lpeek(Lnavar%-460)+2,82 ! 8x8 systemfont header
    Else
        Bmove H6x6+2,Lpeek(Lnavar%-460)+2,82 ! 6x6 systemfont header
    Endif
    Alert 1,"Welk font ?",1,"6x6:8x8:Exit",A
Until A=3
Bmove Varptr(Gemheader$),Lpeek(Lnavar%-460),88 ! en weer terugzetten
Stop
```

## LISTING 4

```
Degas - FDA converter                      Zet hiervoor 'FONT MANIPULATING UTILITY'

Fileselect "%*.FNT", "", Degasnm$
If Degasnm$ <> ""
    Open "I", #1, Degasnm$           ! lees Degas font in
    Degas%=Input$(2048, #1)
    Close #1
    Degas%=Varptr(Degas%)
    Fda$=Space$(4096)                ! ruimte voor nieuw font
    Fda%=Varptr(Fda%)
    Bmove Lpeek(H8x16+76),Fda%,4096 ! kopieer systemfont naar ons font
    Print "Converting..."
    For X%=0 To 127
        For Y%=0 To 15
            Poke Fda%+Y%*256+X%, Peek(Degas%+X%*16+Y%) ! converteren...
        Next Y%
    Next X%
    Lpoke Lpeek(Lnavar%-460)+76,Fda% ! laat eventjes zien
    Fileselect "%*.FDA", "", Fdnam$
    If Fdnam$ <> ""
        Bsave Fdnam$,Fda%,4096
    Endif
    Lpoke Lpeek(Lnavar%-460)+76,Lpeek(H8x16+76) ! en weer terugzetten
Endif
```

## LISTING 5

```
/* FONT MANIPULATING UTILITY Egon Lubbers, '87 */

long h6x6, h8x8, h8x16, lnavar;

get_fntvar()
{
    register long rega0, *regal;

    asm(
        dc.w    0xA000                /* MEGAMAX ALLEEN ! */
        move.l  A0,rega0
        move.l  A1,regal
    )
    lnavar = rega0;
    h6x6 = *regal++;
    h8x8 = *regal++;
    h8x16 = *regal;
}

main()
{
    int i;

    get_fntvar();
    * (long *) (lnavar-22L) = * (long *) (h8x8+76L);
    /* FDA adres veranderen */
    * (int *) (lnavar-46L) = 8;
    /* hoogte karakters */
    * (int *) (lnavar-40L) = 640;
    /* aantal bytes opschuiven */
    * (int *) (lnavar-42L) = 49;
    /* aantal regels */

    for (i = 0; i < 100; ++i)
        printf("Regel %d in beeld\n", i);

    puts("Press RETURN");
    while ((char) gemdos(1) != 13);

    * (long *) (lnavar-22L) = * (long *) (h8x16+76L);
    /* en alles weer herstellen */
    * (int *) (lnavar-46L) = 16;
    * (int *) (lnavar-40L) = 1280;
    * (int *) (lnavar-42L) = 24;
}
```

## LISTING 6

Egon Lubbers' font changing utility (START #6)

Dit programma verandert het 8x16 systemfont in het font waarvan de naam ingetikt (in de desktop: TYP !) moet worden. Font moet in dezelfde directory staan als dit programma

```
fopen equ $3D
fread equ $3F
fclose equ $3E
keep equ $3I

font: bra instal
      ds.b 4096 ; ruimte voor font (slordig)

instal: move.l 4(sp),a3 ; open font
        clr.w -(sp)
        pea 129(a3)
        move #fopen,-(sp)
        trap #1
        addq.l #8,sp
        tst d0
        bmi exit ; error ?
        move d0,d6

        move.l #font,-(sp) ; lees font
        move.l #4096,-(sp)
        move d6,-(sp)
        move #fread,-(sp)
        trap #1
        add.l #12,sp
        tst d0
        bmi exit

        move d6,-(sp) ; close font
        move #fclose,-(sp)
        trap #1
        addq.l #4,sp

dc.w $A000 ; en installeren
move.l -460(a0),a1
move.l #font,76(a1) ; in GEM
move.l #font,-22(a0) ; en in TOS

clr -(sp) ; en bewaar de data
move.l #instal-font+256,-(sp)
move #keep,-(sp)
trap #1

exit: clr -(sp) ; als er iets fout is gegaan...
      trap #1

end
```



# PUBLISHING PARTNER



Dit is een test van een desktop-publishing-programma. Maar niet zomaar een test. Uw recensent werd op slag verliefd op dit programma en van enige objectiviteit is dan ook geen sprake meer. Maar eerst even de zakelijke gegevens. Publishing Partner is een programma van Soft Logic, dat door SoftST uit Amsterdam op de markt wordt gebracht. En om u maar gelijk uit de droom te hel-

pen: dit is een programma om nooit meer mee op te houden, zo mooi. Ben je eenmaal bezig dan zullen je partner en je bed je lange tijd niet zien, behalve die ene partner: Publishing Partner. Het begint al bij het uitpakken van het programma. Een keurig verzorgde handleiding (in het Engels, dat wel) straalt je tegemoet. Een handleiding, ik kan niet anders zeggen, die zeer helder geschreven is. Zo helder, dat ik meteen al dacht: dit is niet door een software-ontwikkelaar geschreven, maar door een professionele tekstschrijver. En dat blijkt ook zo te zijn. Dames en heren software-ontwikkelaars, hier komt u wat van leren. Denk niet: zo'n handleiding kan ik zelf wel in elkaar flansen, want dat kunt u niet. Laat het schrijven aan mensen over die daarvan hun professe hebben gemaakt. Wij schrijvers maken toch ook geen ingewikkelde computerprogramma's; dat laten we met plezier aan u over. Veel gebruikers zullen u dankbaar zijn.

Voor degenen die nog niet weten wat Desktop Publishing eigenlijk is, een kleine uitleg. DP (zo noemen we het voor het gemak maar even, want ik breek mijn vingers over die woorden) is tekstverwerken, illustreren en pagina's opmaken in één. Met zo'n programma en een laserprinter kun je je eigen tijdschrift maken. Nu is dat op zich niet helemaal nieuw, want Apple heeft met zijn Macintosh ook al een configuratie die DP aankant. Maar dat kost u dan (inklusief harddisk en laserprinter) wel zo'n 25.000 gulden. Van Atari kost u hetzelfde (als de laserprinter tenminste snel op de markt komt) nog geen tienduizend gulden.

## LEIDT TOT ECHTSCHEIDING

### TOUR

De handleiding van PP begint met een 'tour' door het programma. De auteur toont ons in tien minuten de meest essentiële en spektakulaire opties van Publishing Partner. De zekerste manier om het programma voorlopig niet meer te verlaten. Voorlopig kunnen nog 'slechts' vier lettertypen worden gebruikt, maar de makers van het programma beloven dat er snel nieuwe uit zullen komen, die dan in een public domain bibliotheek ter beschikking zullen staan. Bravo!

Teksten kunnen zowel in een tekstverwerker worden gemaakt en dan in PP worden 'geïmporteerd' als rechtstreeks in PP worden geschreven. Het laatste heeft het voordeel dat je meteen ziet hoe de tekst eruit komt te zien, maar bij gebruik van een wat kleinere letter (9 punten, de letter die u nu leest) is het op het scherm nog maar moeilijk leesbaar. Wel ziet u precies waar de tekst komt, hoe deze wordt afgebroken, enz.

### WYSIWYG

De bovenstaande tussenkop moet u op z'n Engels uitspreken. Let op: deze term wordt een geveugelde onder de DP-gebruikers. Letterlijk is het de afkorting van 'What You See Is What You Get'. Of, vrij vertaald: zoals het op het scherm komt, wordt de afdruk ook werkelijk. En dat is ook zo. Tekst en illustraties zijn zowel op 'full size'

te zien als vergroot of verkleind. Hoe u maar wilt.

Het werken met het programma is aan de hand van de handleiding kinderspel. Niet dat je snel alle kommando's onder de knie hebt - het zijn er ook zo veel - maar er is

*Uiteindelijk resultaat van Publishing Partner in combinatie met een EPSON FX-80. De letters zijn ratelig en het 'uitvullen' is niet mooi.*

pt

144 points

### Benefits

The benefits of using Publishing Partner are unlimited. By having a word processor, page layout, and forms creator all in one program, you'll be able to quickly and easily create a variety of documents. With Publishing Partner's easy to understand "drop-down" menus, learning and using the program will be instantaneous.

### Special Attributes

Backslant  
Bold  
Double Underline  
Italicize  
Light  
Mirror  
Outline  
Reverse Type  
Shadow  
Strike-through  
Tall  
Underline  
Upside Down  
Wide

You can mix and match any attribute you desire!

For example:

**Tall and Shadow**  
*Italicize, Outline, and underline*

Supported printers include Epson and graphics compatibles, Star (Gemini), Apple LaserWriter and any Postscript compatible device. New print drivers are being released daily so call to make sure your printer is supported.

### Import Graphics



You can reduce, enlarge, or "cut" any Degas or Neochrome picture

### To place an order:

Just call (314) 894-8608 or mail payment to:

SoftLogik Corp  
4129 Old Baumgartner  
St. Louis, MO 63129  
(314) 894-8608



niet begrepen. Inmiddels is er echter een eerste public domain schijfje met printerdrivers en printerfonts in omloop. Het wordt echter (nog?) niet standaard bij het pakket geleverd: de koper moet er zelf naar op jacht.

Het is echt een teleurstelling als je met een eenvoudige matrixprinter gaat afdrukken. Dat is natuurlijk logisch: het programma is bedoeld voor samenwerking met een laserprinter. Ik heb daarover niet de beschikking, zodat ik niet weet hoe het resultaat dan zal zijn. Een 24-naalds printer is waarschijnlijk wel het minste dat je nodig hebt voor een acceptabel resultaat. Gelukkig beschikt PP standaard over de meest gangbare printerdrivers, die zowel 'draft' als 'final' kwaliteit kunnen leveren. Met mijn Star SG-10 zag ik overigens nauwelijks kwaliteitsverschil. In de 'final'-kwaliteit duurde de print echter wel twee keer zo lang. Op pagina 26 kunt u zien hoe het resultaat uit een gewone matrixprinter rolt. Ik moet zeggen: de letterkwaliteit van Signum! ziet er beter uit. Maar dat programma biedt natuurlijk nauwelijks mogelijkheden voor opmaak. Het betekent wel dat er nog wat aan de fonts voor matrixprinters te verbeteren moet zijn. Overigens overweegt Atari de rechten van het programma te kopen en het dan eventueel bij de laserprinter te leveren.

Henri Kruithof

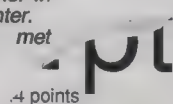
## KONKLUSIE

Zoals u uit het begin van het verhaal al heeft begrepen ben ik enthousiast over Publishing Partner. De prijs (f 499,-) maakt dit programma niet geschikt voor iedereen. Maar voor verenigingen en instellingen, die zelf een verenigingsblad willen maken, is een totale uitgave van zo'n achtduizend gulden (hardwa-

re en software) vaak nog wel te betalen. En dan is het een uitstekend alternatief voor de jaarlijks terugkerende grote uitgaven aan zetter en layouter.

Publishing Partner is een extra reden om de aanschaf van een Atari te overwegen.

*Uiteindelijk resultaat van Publishing Partner in combinatie met de nieuwe Atari Laserprinter. De kwaliteit van tekst en beeld is met sprongen vooruitgegaan het 'uitvullen' blijft te wensen overlaten.*



... on the screen just the way the page will look when it is printed - no more guess work. You can adjust fonts, character sizes, and even character spacing anytime and anywhere on the page. You'll watch an ordinary letter transform into a professional looking "piece" right before your eyes as you experiment with mixing graphics and text.

### Benefits

The benefits of using Publishing Partner are unlimited. By having a word processor, page layout, and forms creator all in one program, you'll be able to quickly and easily create a variety of documents. With Publishing Partner's easy to understand "drop-down" menus, learning and using the program will be instantaneous.

### Special Attributes

Backslant  
Bold  
Double Underline  
Italicize  
Light  
Mirror  
Outline  
Reverse Type  
Shadow  
Strike through  
Tall  
Underline  
Upside Down  
Wide

You can mix and match any attribute you desire!

For example:

**Tall and Shadow**  
Italicize, Outline, and underline

Postscript compatible device. New print drivers are being released daily so call to make sure your printer is supported.

### Import Graphics



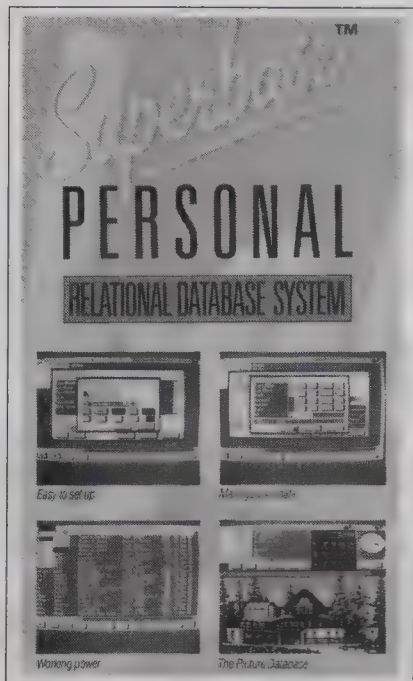
You can reduce, enlarge, or "cut" any Degas™ or Neochrome™ picture

### To place an order:

Just call (314) 894-8608 or mail payment to:

SoftLogik Corp  
4129 Old Baumgartner  
St. Louis, MO 63129  
(314) 894-8608

Produced with Publishing Partner, ATARI ST and a QMS PS 800



f 349,- voor Atari ST, Commodore Amiga en Schneider PC

### Software mogelijkheden

- \* Tot 16.000.000 records - per file
- \* Tot 999 indexen - per file
- \* Maximale file grootte - bepaald door systeem
- \* Aantal files - bepaald door systeem
- \* Aantal open files - bepaald door systeem
- \* Aantal velden per record - bepaald door systeem

Superbase Personal werkt als eerste uit een nieuwe generatie databases met makkelijk te bedienen menu's, meerdere vensters, aanklikpunten en met het voordeel van een relationele database structuur.

### Eenvoudig op te starten

U typt eenvoudig de namen van de velden, voegt er de details aan toe zoals lengte, datum en stijl. U kan de velden verplaatsen over het gehele scherm. U kan zelfs, zonder verstoring van de reeds opgeslagen gegevens, later de volledige layout en formatering wijzigen.

### Beheer van uw gegevens

Superbase visualiseert de gegevens in tabellen of blad voor blad.

Praktisch geen beperking in het aantal velden, met daarenboven volledige controle over wat u uiteindelijk op het scherm wenst te zien.

Selectie van velden en indexen, aanwending van VCR controles voor visualisatie van uw gegevens, fast forward, rewind, pause, stop, hetzelfde als met uw cassette recorder. Een uniek filtersysteem laat selectie en verwerking toe met om het even welke soort records in uw bestanden.

### Werkvermogen

Definieer rapporten en selecties in diverse bestanden, met verschillende sorteerniveaus. Voer gegevens in van andere bestanden, voer gegevens uit naar uw tekstverwerker of verenig verschillende bestanden in een nieuwe database.

De B+ boomstructuur en disk buffering laten enorme mogelijkheden toe. Superbase leest een naam en adres record in minder dan drie honderste van een seconde.

### De plaatjes database

Superbase laat toe beelden, grafieken en tekeningen te linken aan uw records. U krijgt aldus een uniek systeem van beelden archivering. U kan zelfs een automatische "slide show" genereren.



Precision  
Software

## Softst BV

Nw. Prinsengracht 75  
1018 VR Amsterdam  
020-236393

### Data

- \* Tekst, datum, numerieke & externe
- \* Input controle
- \* Volledig te definiëren velden
- \* Kalender 1-9999 A.D., meerdere mog.
- \* 13-cijferige nauwkeurigheid
- \* Kodewoord beveiliging

### Rapporten

- \* Tot 255 tekens per regel
- \* Titel, datum, pagina nummer
- \* Telt records, subtotalen en totalen
- \* Meerdere files kunnen naar scherm, pr., etc.
- \* Multi kolom, variabele labelformaten
- \* Sorteert in elk formaat



# NIEUW VAN P.D.S.

nu verkrijgbaar bij uw Atari-dealer

## GRAFISCH TABLET

ideaal voor tekenaar en architect  
prijs inclusief software

f 1198,—

## VIDEO MIXER (GENLOCK INTERFACE)

pakket waardoor uw Atari ST 1040 computerbeelden  
kan mengen met normale TV-beelden

## STEVE

razendsnelle database met grafische mogelijkheden  
volledig in 68000 geschreven

f 438,—

## STAD

professioneel tekenpakket met extra  
3-D gedeelte, telecommunicatie en scanner

f 239,—

## IN ONTWIKKELING BIJ P.D.S.

(raadpleeg uw dealer - wellicht zijn enkele  
pakketten nu reeds leverbaar)

## SIGNUM! (Nederlandse versie)

## PRO SPRITE DESIGNER

sprite editor met ingebouwde mogelijkheid om  
animaties te testen

## PRO SOUND DESIGNER

sound sampler compleet met hardware

### VERDER IS VERKRIJGBAAR:

#### PROGRAMMEERTALEN

Megamax C-compiler	f 725,—
Megamax B-Tree Toolbox	f 249,—
Modula-2/ST 2.0	f 449,—
Systeem Development Kit	f 695,—
K-Seka	f 249,—
K-Resource ST	f 199,—
GST-C	f 285,—
GST Fractal Generator	f 79,—
GST Macro Assembler	f 189,—
Metacomco Assembler	f 249,—
BCPL Compiler	f 295,—
Lattice C	f 449,—
Cambridge Lisp	f 395,—
Metacomco Pascal	f 295,—
Metacomco Make	f 199,—
Pro Fortran 77	f 595,—
Pro Pascal	f 595,—

#### ADMINISTRATIEVE SOFTWARE

ST Professional Module 1	f 714,—
ST Professional Module 1 + 2	f 1314,—
ST Professional Module 1 + 3	f 1674,—
ST Professional Update M2	f 714,—
ST Professional Update M3	f 1074,—

#### CAD/CAM

Easy-Draw2	f 295,—
The Graphic Artist	f 995,—

#### DATABASES

Habaview	f 149,—
Laser Base	f 225,—
Trimbase	f 295,—
Habadex	f 95,—

#### SPREADSHEETS

K-Spread 1.2b	f 199,—
K-Graph	f 199,—
Vip Professional (tekst)	f 795,—
Vip Professional (GEM)	f 895,—

#### TEKSTVERWERKING

Signum!	f 539,—
Habawriter (Nederlands)	f 195,—
Habamerge	f 95,—

#### BEELD & ANIMATIE

Artdirector	f 195,—
Filmdirector	f 225,—
Paintworks	f 119,—
Maps & Legends	f 99,—
Degas Elite	f 189,—
Art Gallery	f 99,—
Art Gallery II	f 99,—
Art Gallery III	f 99,—
Print Master	f 125,—

#### MIDI MUZIEK

MidiPlay	f 149,—
Musidisk - Klassiek I	f 59,—
Musidisk - Klassiek II	f 59,—
Musidisk - Beatles	f 59,—

#### UTILITIES

Demo Constructor	f 495,—
Desk Assist II	f 199,—
LQ-Font	f 169,—
Multi-HardCopy	f 139,—
ST Key	f 99,—
ST Toolkit	f 99,—
K-Comm	f 179,—
K-Ram	f 99,—
K-Switch	f 119,—
VT 100 Terminal Emulator	f 199,—
Aladin Macintosh Emulator	f 375,—
Mac Emulator	f 495,—
IBM Emulator	f 295,—
Hardware Clock	f 95,—

(prijswijzigingen voorbehouden)

# PDS

KWALITEITS SOFTWARE EN GOEDE SERVICE BEL VOOR INFORMATIE 070 - 999 039



# 'S P E L L B O U N D

## Universele Spellingschecker

*Er zijn tekstverwerkers te over voor de ST. Een tekstverwerker met een goed Nederlands woordenboek is er echter nog niet. Als die er komt is het nog maar de vraag of de tekstverwerker ook op andere punten dermate krachtig is dat u uw huidige programma ervoor terzijde legt.*

*Daarom vroeg START aan Rob Koopman of hij een spellingschecker wilde schrijven die met elke tekstverwerker te gebruiken is.*

*Dat leidde tot SPECIAL DISK 4: SPELLBOUND.*

*In het volgende artikel legt de auteur uit hoe het programma werkt. Mocht u zelf een spellingschecker willen schrijven, dan geeft dit artikel u alle benodigde handvaten. De gepresenteerde aanpak is echter ook geschikt om andere problemen op te lossen.*

### GRAMMATIKA

Een spellingschecker is een prachtig hulpmiddel voor iedereen die regelmatig via een computer artikelen of brieven produceert. Het hulpmiddel controleert echter alleen de spelling van een woord. "Ik wort" levert beslist een foutmelding op. Grammatikale blunders komt zo'n programma echter niet op het spoor. "Ik wordt" is voor een spellingschecker korrekt Nederlands.

Misschien heeft u zich wel eens afgevraagd hoe een programma dat de spelling van een tekst controleert in elkaar zit. Uiteraard heeft zo'n programma een uitgebreide woordenlijst nodig. Die woordenlijst moet zo compact mogelijk worden opgeslagen. Daarna zal een procedure in korte tijd uitsluitend moeten geven of een woord al dan niet in de woordenlijst voorkomt.

Het maken van een lijst met woorden die wel in een tekst maar niet in het woordenboek voorkomen is een eenvoudige taak. De computer moet alle woorden van de te onderzoeken tekst eerst alfabetisch sorteren. Daarna moet hij de gesorteerde lijst vergelijken met het woordenboek. Alle woorden die niet in het woordenboek voorkomen, worden afgedrukt.

Op zich is dit een prima methode. Het sorteren van de woorden in een tekst is van de orde  $N \log N$  ( $N$  is het aantal woorden in de tekst). Het vergelijken van de tekst met het woordenboek is van de orde  $N+M$ . ( $M$  is het aantal woorden in het woordenboek). Zolang de te onderzoeken tekst in RAM gesorteerd kan worden, zal de procedure vrij weinig tijd kosten. Het grote voordeel is dat de woorden in het woordenboek ieder maar één keer bekeken worden. Het programma kan het woordenboek dus sequentieel van schijf lezen. Het grote nadeel van deze methode is dat het opzoeken van slechts één woord een paar seconden duurt. Het hele woordenboek moet immers worden doorgekeken tot het gezochte woord gevonden is.

### BINAIR ZOEKEN

Ik heb onderzocht of het mogelijk is een spellingschecker te maken die een ingetikt woord direkt controleert. Het sequentieel doorzoeken van het woordenboek duurt dan echter veel te lang. Een oplossing is zoeken door "halveren". Een woord uit de tekst wordt eerst vergeleken met het woord dat in het midden van het woordenboek

staat. Als het woord achter het "middelste" woord thuis hoort, wordt het midden van de tweede helft van het woordenboek bekeken. In het andere geval het midden van de eerste helft. Dit proces herhaalt zich tot het interval nog maar één woord groot is. Op die manier kan in maximaal zestien vergelijkingen een woord in een woordenboek van 65.000 woorden gevonden worden.

Het lezen van een woord van een hard disk kost ongeveer 1/20 seconde. Het opzoeken van een woord kost dus in het ongunstigste geval zestien keer zolang: ongeveer driekwart seconde. Dit is onaanvaardbaar lang. Het woordenboek moet dus in het geheugen worden opgeslagen. Op een Atari ST met 1 megabyte geheugen is dit heel goed mogelijk. Een woordenboek met 60.000 woorden zal ongeveer een half megabyte in beslag nemen. Het opzoeken van een woord gaat dan vrij snel.

Natuurlijk is een woordenboek van 500.000 bytes onhandelbaar groot. Met een 520 ST komen we bovendien geheugen tekort. Er zijn gelukkig wat truuks om het woordenboek kleiner te maken. Zo kunnen bijvoorbeeld drie letters in twee bytes worden opgeslagen. Een letter kost vijf bits, drie maal vijf bits maakt vijftien bits en dat is kleiner dan twee bytes (zestien bits). Verder kunnen bepaalde veel voorkomende standaarduitgangen en prefixen met een code aangegeven worden. Al met al is het waarschijnlijk mogelijk het woordenboek tot ongeveer 250.000 bytes terug te brengen. Dit is echter nog steeds te groot. Bovendien maakt iedere datareduktietechniek het zoeken ingewikkelder en dus dreigt de zoektijd langer te worden.

### DUBBELGANGERS

Het kijken of de woorden van een tekst in een woordenboek voorkomen, lijkt heel erg op het volgende probleem: stel we hebben een lijst van vijfhonderd getallen onder de duizend. Hoe kunnen we zo efficiënt mogelijk kijken of er dubbelen in de lijst voorkomen?

Op het eerste gezicht lijkt de beste oplossing de lijst eerst te sorteren. Daarna kan hij eenvoudig op dubbeltallen worden gecontroleerd. Een veel snellere oplossing is echter gebruik te maken van een array van 1000 bits (= 125 bytes). Eerst wordt de hele array op nul gezet. Daarna onder-

zoekt de procedure elk getal. Is het getal bijvoorbeeld 756, dan kijkt het of bit 756 op één staat. Is dat het geval, dan is 756 dubbel. Staat bit 756 nog op nul dan wordt het bit nu op één gezet.

In pseudocode:

```
procedure toondubbel (lijst);
verzameling = set of 0..999;
begin
  verzameling := [];
  for alle getallen x in de lijst do
    if x in verzameling then
      dubbel
    else
      incl (x,verzameling)
    end;
  end;
end;
```

Op dezelfde manier kan geprobeerd worden een woordenlijst op te slaan. We verzinnen daarvoor een functie die een woord omvormt tot een uniek getal. In plaats van de woorden kunnen we dan de getallen in een woordenlijst zetten. Er is maar een beperkt aantal Nederlandse woorden. Bij een perfecte functie zouden we dus aan de getallen 1 tot en met 1.000.000 meer dan genoeg hebben.

Jammer genoeg bestaat deze functie niet. Als de computer echter af en toe een foutje mag maken, kunnen we deze functie wel benaderen.

Een functie die aan een woord een getal toekent, wordt in de literatuur ook wel hashfunctie genoemd (door-elkaar-hak-functie).

Als we de Nederlandse woorden afbeelden op de getallen 1 t/m 1 miljard is de kans dat twee woorden hetzelfde getal toebedeeld krijgen vrij klein. Op die manier is het mogelijk elk Nederlands woord in een code van 32 bits te veranderen. Een woordenboek met 60.000 woorden kan dan in 240 kbytes opgeslagen worden. Helaas is het bij deze methode niet meer mogelijk om van de hashwaarde terug te gaan naar het bijbehorende woord.

Om te bepalen of een woord in de woordenlijst voorkomt, wordt eerst de hashfunctie op het woord losgelaten. Daarna kijkt het programma of de aldus verkregen waarde in het "getallenboek" voorkomt.

Het is in principe mogelijk dat een woord fout gespeld is, terwijl zijn hashwaarde toch in het woordenboek zit. Een korrekt gespeld woord heeft dan toevallig dezelfde hashwaarde. Bij een woordenlijst van 60.000 woorden is de kans hierop ongeveer 1 op 16.000 (1 giga / 60k).

Het voordeel van de hashmethode is dat de woordenlijst klein is en dat er bijzonder snel gezocht kan worden. Het nadeel is dat er af en toe een foutje doorslipt en dat niet meer te zien is welke woorden in de lijst staan.

Zelf vind ik het aardige dat de stelling "beter zonder fouten dan snel" hier niet op blijkt te gaan.

Overigens gebruikt de politie een dergelijke methode al langer. In plaats van een kopie van een persoon wordt naar vingerafdrukken gekeken. Theoretisch kan dat natuurlijk ook fout gaan.



## VERDERE VERFIJNINGEN

Als eenmaal de stap is gemaakt om in plaats van woorden hashwaarden op te slaan, zijn er weer nieuwe mogelijkheden om het getallen-woordenboek efficiënter op te slaan.

Eerlijk gezegd is vier bytes per woord nog veel te veel. Een echt goed compact woordenboek mag niet veel meer dan twee bytes per woord beslaan. Om dat te bereiken staat een aantal wegen open.

1) De programmeur kan meer dan één hashfunctie gebruiken.

Als een woordenboek 60.000 woorden moet bevatten en als twee bytes per woord beschikbaar zijn, zou het woordenboek kunnen worden opgeslagen in een rij van 1.000.000 bits. Deze rij beslaat 125.000 bytes in het geheugen. Als nu gebruik wordt gemaakt van een hashfunctie die waarden tussen 1 en 1.000.000 genereert, kan ieder woord gerepresenteerd worden door een bit in de rij.

Dit is echter niet goed genoeg voor een spellingswoordenboek van enige kwaliteit. De kans dat een fout woord ten onrechte goed wordt gekeurd is ongeveer 1 op 17. We kunnen echter ook op ieder woord twee verschillende hashfuncties loslaten. Dit levert waarschijnlijk twee verschillende getallen op. Deze twee getallen slaan we op als bits in de rij. Alleen woorden waarvan beide hashwaarden in de rij voorkomen, worden goedgekeurd. De rij is bij gebruik van twee hashwaarden voor 12% gevuld. De kans dat een fout woord toch twee hashwaarden heeft die in het woordenboek (de rij) voorkomen is dus 1.44%. Natuurlijk kunnen we ook nog meer hashfuncties gebruiken. De bijgevoegde tabel geeft een overzicht van de korresponderende foutkansen.

aantal hash- functies	1.000.000 bits fout in %	1.200.000 bits fout in %
1	6.0	5.0
2	1.4	1.0
3	0.6	0.3
4	0.3	0.2
5	0.2	0.1
6	0.2	0.07
7	0.2	0.06
8	0.3	0.07
9	0.4	0.08
10	0.6	0.1

(alle getallen op basis van 60.000 woorden)

Bij zes hashwaarden wordt dus ongeveer één op de vijfhonderd foute woorden ten onrechte goed gerekend. Als we 20% meer bits gebruiken, wordt bij zeven hashfuncties ongeveer 1 op de 1.400 foute woorden ten onrechte goed gerekend. Hoe groter het aantal getallen waar we de Nederlandse woorden op afbeelden, hoe kleiner de kans op fouten.

Als gemiddeld in een document twintig spelfouten voorkomen, blijft in het eerste geval maar in 1 op de 25 documenten een fout zitten. Dit lijkt mij aanvaardbaar. Het nadeel van deze methode is wel dat het verantwoord verwijderen van woorden vrijwel onmogelijk is geworden.

2) De programmeur kan in plaats van de getallen de afstanden tussen de getallen opslaan.

Wie accepteert dat bijvoorbeeld 1 op de 100 spelfouten niet gevonden wordt, kan in plaats van de hashwaarden zelf de af-

standen tussen de waarden opslaan. Deze getallen liggen dan gemiddeld ongeveer 100 posities van elkaar af.

Als de afstanden tussen de bytes met behulp van een variabele lengte codering worden weergegeven, zal het woordenboek ongeveer twaalf bits per woord beslaan (Dit is natte vinger werk. Ik heb het niet uitgerekend, maar een beetje meer of minder maakt niet veel uit.) Het totale woordenboek is dan maar 90.000 bytes groot. Als de afstand tussen twee getallen kleiner is dan 128, wordt de afstand met een byte weergegeven. Is de afstand groter, dan wordt bit zeven van het byte op één gezet en een tweede byte toegevoegd, enz. Als 1 op 100 niet acceptabel is, kan de gemiddelde afstand tussen de hashwaarden natuurlijk groter gemaakt worden.

3) De programmeur kan de hashwaarden gehashed opslaan.

Een derde methode is het gehashed opslaan van de uit de woorden berekende hashwaarden. Op het eerste gezicht levert dat niets op, maar er is een truuk beschikbaar. Normaal houdt hashen in dat het geheugen wordt ingedeeld in een aantal 'buckets' (bakjes?). Door middel van een hashfunctie wordt berekend in welk bakje een woord wordt opgeslagen. Het systeem lijkt een beetje op het sorteren van brieven via de postkode.

'Pseudo' definitie hashtable:

type bucket = array [0..bucketsize] of woorden;  
hashtable = array [0..maxbucket] of bucket;

Normaal slaan we bij zo'n aanpak de woorden in de buckets op. Het is echter ook mogelijk een tweede hashfunctie op de woorden los te laten. In plaats van het woord wordt dan de hashwaarde opgeslagen. Aangezien ook al de bucket waarin het woord terecht komt via hashing bepaald wordt, kan met twee bytes per woord worden volstaan. Als bijvoorbeeld 8096 buckets ter beschikking staan, zijn er ongeveer 500 miljoen mogelijke combinaties. De kans dat bij een woordenboek van 60.000 woorden een fout woord toch in het woordenboek gevonden wordt, is ongeveer 1 op 8.000 (= het aantal buckets).

De totale grootte van het woordenboek is ongeveer 140.000 bytes (aantal woorden maal twee plus het aantal buckets maal twee). Deze laatste methode is toegepast in het programma SPELLBOUND.

## LISTING

Ik geloof zelf niet zo erg in het letterlijk overnemen van programma's uit tijdschriften. Het is veel leerzamer de aanpak van anderen te bestuderen om daarna de goede dingen over te nemen in een eigen programma. Liefst met een heel andere toepassing. De techniek die hier gebruikt wordt om compact op te slaan, hoeft immers niet beperkt te blijven tot woorden. Op dezelfde manier kan bijvoorbeeld een schaakprogramma stellingen uit openingen herkennen.

De bijgevoegde listing bevat het hart van de spellingschecker: de hashfunctie en de opslagmethode. Ik hoop dat de programmeur duidelijk genoeg is om ook mensen die niet in Modula-2 thuis zijn op het goede spoor te zetten.

### Literatuur

• *Programming pearls* / Jon Bentley.

Addison-Wesley / ISBN 0-201-10331-1

• *Data structures using Modula-2*

R.F. Sincovec / R.S. Wiener.

John Wiley / ISBN 0-471-81489-X

## SPELLBOUND

Special disk viert bestaat uit vijf delen. Allereerst de eigenlijke spellingschecker (SPELTEST) en het Nederlandse woordenboek (GROEN.WRD). Het 'groene boekje' bestaat uit twee delen. Een bitarray van 27<sup>3</sup> bytes voor woorden kleiner dan vier letters en een hashstructuur (zie artikel) voor langere woorden. Nadat u met uw favoriete tekstverwerker een tekst gemaakt heeft, kiest u SPELTEST en biedt uw bestand ter beoordeling aan. De spellingschecker toont daarop alle onbekende en fout gespelde woorden op het scherm.

De snelheid van het programma is onafhankelijk van het aantal spelfouten en bedraagt meer dan duizend woorden per seconde. Aan de hand van de foutmeldingen kunt u uw artikelen en brieven gemakkelijk corrigeren. De snelheid van SPELLBOUND is dermate hoog dat de controle beslist geen hinderlijk onderdeel van het schrijfsproces wordt. Integendeel: het is juist leuk om te zien hoe de kwaliteit van uw teksten onder invloed van de vriendelijk op de vingers tikkende ST met sprongen vooruit gaat.

Programma plus woordenboek worden in dertig seconden in het geheugen geladen. Een behoorlijke tekst (15 K) controleren, kost daarna slechts vijf seconden. Wie SPELTEST vanaf een harddisk laadt, is zelfs al in tien seconden met het hele proces van laden en controleren klaar. Het derde deel van het pakket, BLINK, controleert woorden terwijl u ze tikt. Zo snel BLINK een woord niet herkent, zal het dat woord een korte tijd laten knippen.

BLINK werkt samen met alle ons bekende tekstverwerkers. Het programma leest de toetsenbordbuffer via een interruptroutine. Theoretisch is het mogelijk dat een ander programma iets dergelijks doet, waardoor problemen ontstaan. We zijn zoiets gelukkig nog niet tegengekomen.

Uiteraard zal het programma af en toe ook goed gespelde woorden ter correctie aanbieden. Vooral lange samengestelde woorden, leenwoorden en 'vakjargon' kent het woordenboek niet.

Wie extra woorden aan het woordenboek wil toevoegen, wordt op zijn wenken bediend door het programma SPELMAAK. En mocht u er op een bepaald moment achter komen dat een woord ten onrechte in het woordenboek is gezet, dan kunt u het verwijderen met SPELKILL.

Het is ook mogelijk om zelf woordenboeken voor andere talen te maken. Dat kan Engels zijn, maar bijvoorbeeld ook de verzameling gereserveerde woorden van een programmeertaal. BLINK beperkt de samenwerking namelijk niet alleen tot tekstverwerkers..!

Het programma SPELMAAK produceert automatisch een bestand GROEN.WRD als het dit nog niet op een schijf vindt. In principe bouwt u het woordenboek op door foutloos gespelde teksten aan te bieden. Een complete woordenlijst versnelt het proces natuurlijk aanzienlijk.

Abonnees kunnen SPELLBOUND voor de aantrekkelijke prijs van f 49,- bestellen. Niet-abonnees betalen f 69,-. Zie onze rubriek "lezersservice" voor de bestelwijze.



# IMPLEMENTATION MODULE GroeneBoekje;

```

ROUTINES voor het creëren en raadplegen van een woordenlijst voor een
spellingschecker. Deze tekst is een vereenvoudiging van de echte source.
Onderstaande routines worden alleen gebruikt bij het aanmaken van het
woordenboek. De spellingschecker gebruikt een compacte versie van het
woordenboek, waarbij lege buckets verwijderd zijn.
auteur: r.koopman
datum: 28 april 1987

```

```

CONST
    tabelSize = 27*27*27; (* aantal woorden van minder dan 4 letters *)
    tabelSize = 8191; (* aantal buckets *)
    bucketsize = 15; (* maximale aantal elementen in bucket *)
TYPE
    (* Datastructuur die bij de opbouw van het woordenboek gebruikt wordt. *)
    bucket =
        RECORD
            i : CARDINAL; (* lengte bucket *)
            c : ARRAY [1..bucketsize] OF CARDINAL; (* bucket zelf *)
        END;
    kleintabel = SET OF [0..tabelSize];
    groottabel = ARRAY [1..tabelSize] OF bucket;

```

```

VAR
    klein : kleintabel; (* alleen in gebruik bij opbouw *)
    groot : groottabel; (* van het woordenboek *)
    (*
    Dit is een schets van de routines die gebruikt kunnen worden voor
    het konstrueren van het woordenboek. Er is schaaateloos gebruik gemaakt
    van de geheugenruimte van de Atari. Deze oevang is gekozen om de kans
    op 'dubbelgangers' te minimaliseren.
    Het woordenboek kan maximaal 60.000 woorden bevatten.
    *)

```

```

PROCEDURE InitieerTabellen;
VAR
    i : CARDINAL;
BEGIN
    (* maak verzameling van woorden < 4 letters leeg *)
    klein := kleintabel();
    (* maak buckets leeg door lengtes op nul te zetten *)
    FOR i:=1 TO tabelSize DO
        groot[i].l := 0;
    END;
END InitieerTabellen;

```

```

PROCEDURE EnterWoord (VAR w:ARRAY OF CHAR; w1:CARDINAL);
(*
Deze routine zet een woord w bestaande uit hoofdletters met lengte w1
zonodig in het woordenboek.
*)

```

```

VAR
    i,hash,buck : CARDINAL;
BEGIN
    (* bepaal hash waarde en bucket nummer *)
    HashValues (w,w1,hash,buck);
    (* zet korte woorden in de kleine tabel en grote in de grote tabel *)
    IF w1<4 THEN
        INCL (klein,hash);
    ELSE
        WITH groot[buck] DO
            (* zoek woord in woordenboek *)
            FOR i:=1 TO l DO
                IF c[i] = hash THEN
                    RETURN (* woord is al aanwezig *)
                END;
            END;
        END;
    END;

```

```

END;
END;
(* zet als er nog ruimte is het woord in het woordenboek *)
IF l<bucketsize THEN l:=l+1; c[l]:=hash; END;
END;
END;
END EnterWoord;

```

```

PROCEDURE TestWoord (VAR w:ARRAY OF CHAR; w1:CARDINAL) : BOOLEAN;
(*
Deze routine is eigenlijk overbodig. Hij maakt alleen het grote
woordenboek compleet. De routine is praktisch identiek aan EnterWoord.
TestWoord=TRUE => w is element van woordenlijst
*)

```

```

VAR
    i,hash,buck : CARDINAL;
BEGIN
    HashValues (w,w1,hash,buck);
    IF w1<4 THEN
        RETURN hash IN klein;
    ELSE
        HashValues (w,w1,hash,buck);
        WITH groot[buck] DO
            FOR i:=1 TO l DO
                IF c[i] = hash THEN RETURN TRUE END;
            END;
        END;
        RETURN FALSE;
    END;
END TestWoord;

```

```

PROCEDURE HashValues (VAR w:ARRAY OF CHAR;w1: CARDINAL; VAR h1,h2:CARDINAL);
(*
De twee hashfuncties zijn op het oog gekozen. Ik sta niet garant voor de
kwaliteit. Voor het gemak wordt hier de overflow test uitgeschakeld.
Dit is natuurlijk niet erg portable (de meeste processoren zullen een
Trap genereren).
Bij woorden van minder dan 4 letters wordt in h1 geen hashwaarde, maar
een eenduidig getal geschreven tussen 1 en 27^3.
*)

```

```

CONST
    c1 = 31415; c2 = 947;
VAR
    i : CARDINAL;
    h : LONGCARD;
BEGIN
    IF w1<4 THEN
        h1 := 0;
        FOR i:=0 TO w1-1 DO
            h1 := h1*c1 + ORD(w[i]) - ORD('0');
            h2 := (h2*c2 + ORD(w[i]) - ORD('0'));
        END;
        h2 := h2 MOD tabelSize + 1; (* h2 = bucket nummer *)
        (**= zet compiler tests weer terug naar oude toestand *)
    END;
    END HashValues;
END GroeneBoekje.

```



# HET GEMAK DIENT DE MENS: ACCLOAD

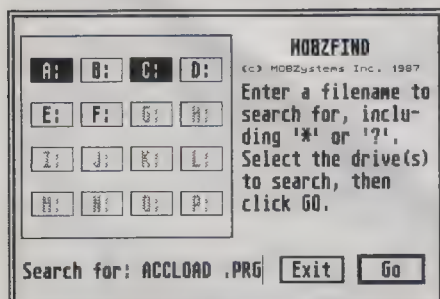
Deskaccessoires zijn programma's die bij het booten van de computer worden ingeladen. Er zijn hele leuke en handige accessoires bij en men is snel in de verleiding om er (te) veel op de boot-schijf te zetten. Vooral als er grotere utilities bij zitten, zoals sideclick, duurt het booten erg lang. Bovendien nemen de accessoires een flink stuk geheugen in beslag. De gebruiker moet dus een evenwicht tussen het aantal accessoires enerzijds en geheugenruimte en inlaadtijd anderzijds zien te vinden.

Een ander probleem is dat er maximaal zes accessoires ingeladen mogen worden. Dit betekent dat veel gebruikers meerdere bootschijven hanteren.

Al dat getob is nu verleden tijd: Wim de Winter vond een manier om alle accessoires op een bootschijf te laten staan en er tijdens een reset maximaal zes uit te kiezen. ACCLOAD is geschikt voor gebruik in alle resoluties.

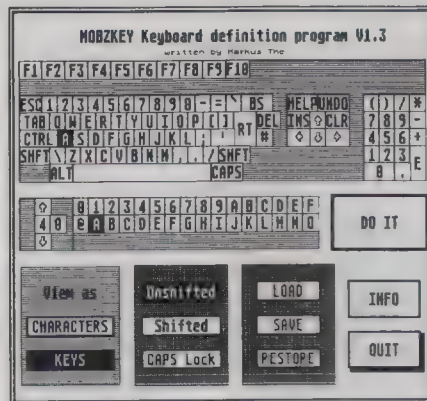
## ACCESSOIRE BIBLIOTHEEK

Het programma ACCLOAD geeft een lijst van alle ACC's en geïnnactiveerde ACC's (kenbaar aan de extensie ACX). De gebruiker heeft de mogelijkheid de extensie zo te veranderen, dat de accessoire wel of juist niet wordt ingeladen. Op die manier kan een bibliotheek van maximaal 25 ACC's op de bootschijf staan.



Het in- en uitschakelen van de accessoires gebeurt met de toetsen die gewoonlijk de cursor besturen. De plaats waar je bent is herkenbaar doordat deze in diapositief is weergegeven. Met "op" en "neer" wandel je door de lijst. Met "rechts" schakel je een accessoire uit, met "links" aktiveer je hem. Dit kommando-overzicht

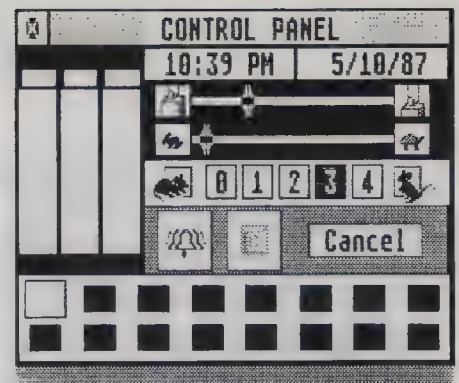
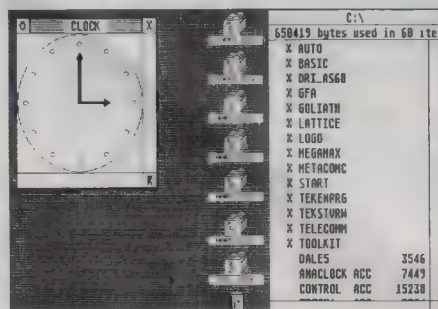
komt ook op het scherm wanneer de toets "Help" wordt ingedrukt. "Undo" geeft in geval van problemen de beginsituatie weer terug. Als de gewenste selectie bereikt is, moet er op "return" gedrukt worden. De gekozen accessoires worden ingeladen en het booten verloopt verder als gebruikelijk. Als u tijdens het laden van ACCLOAD op "return" drukt (het programma geeft zelf het gewenste moment aan), wordt de routine afgebroken. U krijgt dan de accessoires die u de vorige keer actief had gemaakt.



## COMPILEREN

Het programma ACCLOAD werkt alleen als het in een folder met de naam AUTO op de bootschijf staat. Dit stelt enige eisen aan het programma. Op het moment dat het TOS de AUTO-folder raadpleegt, is er namelijk nog geen GEM aanwezig. GfA BASIC maakt gebruik van GEM. ACCLOAD kunnen we dus niet als BASIC-bestand in de AUTO-folder plaatsen. Maar er is een oplossing: de GfA compiler. Deze produceert een op zichzelf staand programma dat alleen van GEM gebruik maakt als er instructies in het programma voorkomen die GEM aanroepen. In ACCLOAD heb ik dat niet gedaan en dus loopt het gecompileerde programma onder TOS.

ACCLOAD is gecompileerd met versie 1.6 van de GfA-compiler. Mocht u die niet bezitten: op START-schijf zes staat het programma zowel als listing als gebruiksklaar. Bovendien is de file ACCLOAD.DAT toegevoegd, die voor een wat mooiere layout zorgt. De mogelijkheden van AES (een onderdeel van GEM), zijn immers nog niet aanwezig. Het programma werkt ech-

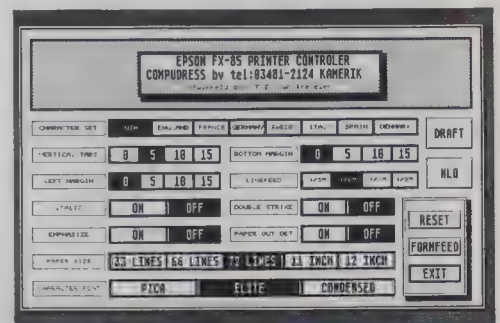


ter ook zonder deze file. ACCLOAD.DAT moet overigens eveneens in de AUTO-folder staan.

## PROGRAMMABESCHRIJVING

Voor het inlezen van de directory maakt het programma gebruik van de functie BIOS(4). De hoofddirectory bevindt zich altijd op zeven sectoren, die op schijf te vinden zijn vanaf sektor nummer elf. Deze twee parameters worden aan het BIOS doorgegeven, dat de inhoud van de directory in een string plaatst. Elke directory entry neemt hierin exakt 32 bytes in. Begint een entry met chr\$(0), dan staan er verder geen bestanden meer in de directory.

De bestandsnamen worden vervolgens uit deze 32-byte substrings gehaald en in een aantal arrays gezet. Via wat scherm- en toetsenbord communicatie kan de gebruiker bepalen welke filenames moeten veranderen. De wijzigingen worden tenslotte met de GfA-functie "Name" op de schijf aangebracht. Daarna vervolgt het boot-proces op de normale manier.



## VOLGORDE

Als het programma GDOS ook in de autofolder staat, moet u er op letten, dat ACCLOAD het eerst in werking wordt gesteld. Gebeurt dat niet, dan moet u het volgende doen: via de desktop (file-show info) geeft u GDOS.PR de naam GDOS.PR. Eenmaal "backspace" en een "return" zijn daarvoor genoeg. Vervolgens kopieert u GDOS.PR in de autofolder. Hierdoor ontstaat natuurlijk een "naamconflict". Daarom maakt u de naam van de kopie GDOS.PR.G. Na afloop gooit u GDOS.PR weg. Hierdoor is ACCLOAD op de eerste plaats in de AUTO-folder komen te staan.



```

*****
* ACC-LOAD - Accessory Loader *
* Auteur      Wim de Winter *
*****

Neg$=Chr$(27)+"p"
Pos$=Chr$(27)+"q"
Up$=Chr$(27)+"A"
Dwn$=Chr$(27)+"B"
Resolutie%=Xbios(4)

@Prohibit
@Init
@Lees
@Selecteer_op_aktiviteit
@Interaktie

Procedure Init
  Chdir "a:\Auto"
  Dim Aktief!(25),Is_veranderd!(25)
  Dim Acc$(25),Acsave1$(25),Acsave0$(25),Accundo$(25)
  Nohelp$=String$(32000,Chr$(255))
  Sput Nohelp$
  If Resolutie%=2
    @Vulbox(252,10,610,250,0)
    @Vulbox(253,11,609,249,1)
    If Exist("AccLoad.Dat")
      Open "R",#1,"acclload.dat",2910
      Field #1,2910 As B$
      Get #1
      Put 262,20,B$
      Get #1
      Put 262,250,B$
    Endif
    Print At(46,1);"Typ Help voor hulp"
  Else
    Print At(11,25);"Typ Help voor hu =>"
  Endif
  Sget Nohelp$
  Return

Procedure Selecteer_op_aktiviteit
  For IX=1 To Aantal_accs%
    If Right$(Acc$(IX),2)="CC" Then
      Aktief!(IX)=True
      Acc$(IX)=Acc$(IX)+" " +Space$(12)
      Inc Aantal_aktief%
    Else
      Aktief!(IX)=False
      Acc$(IX)=Space$(12)+" " +Acc$(IX)
    Endif
  Next IX
  Return

Procedure Interaktie
  @Lijst
  Lijn%=1
  Do
    IX=Inp(2)
    Print Pos$;Acc$(Lijn%)
    Print Up$;
    If IX=13
      @Save_changes
      Endif
    If IX=200
      Print Up$;
      Endif
    If IX=203
      @Aktiveer(Lijn%)
      Endif
    If IX=205
      @Inaktiveer(Lijn%)
      Endif
    If IX=208 And Lijn%<Aantal_accs% =>
      ! v
      Print Dwn$;
      Endif
    If IX=225
      ! Undo
      D.name$=D.save$
      Arrayfill Is_veranderd!(),False
      Clr Aantal_aktief%
      For JX=1 To Aantal_accs%
        Acc$(JX)=Accundo$(JX)
        Acsave1$(JX)=Acsave0$(JX)
      Next JX
      @Selecteer_op_aktiviteit
      @Lijst
    Endif
    If IX=226
      @Help(Lijn%)
      Endif
    Lijn%=Crslin
    Print Neg$;Acc$(Lijn%)
    Print Up$;
  Loop
  Return

Procedure Lijst
  Local IX
  Sput Nohelp$
  Print At(1,1);

```

```

  For IX=1 To Aantal_accs%
    Print Acc$(IX)
  Next IX
  Print At(1,1);Neg$;Acc$(1)
  Print At(1,1);
  N=1
  Return

Procedure Aktiveer(NX)
  If Not Aktief!(NX)
    Acsave1$(NX)=Acsave1$(NX)+"C"
    Acc$(NX)=Space$(12)
    Rset Acc$(NX)=Acsave1$(NX)
    Acc$(NX)=Acc$(NX)+" " +Space$(12)
    Aktief!(NX)=True
    @Onthoud_verandering(NX)
    Inc Aantal_aktief%
  Endif
  Return

Procedure Inaktiveer(NX)
  If Aktief!(NX)
    Acsave1$(NX)=Left$(Acsave1$(NX),I =>
nstr(Acsave1$(NX),"")*2)
    Rset Acc$(NX)=Acsave1$(NX)
    Mid$(Acc$(NX),14,1)="!"
    Aktief!(NX)=False
    @Onthoud_verandering(NX)
    Dec Aantal_aktief%
  Endif
  Return

Procedure Onthoud_verandering(NX)
  If Not Is_veranderd!(NX) Then
    Is_veranderd!(NX)=True
  Else
    Is_veranderd!(NX)=False
  Endif
  Return

Procedure Save_changes
  If Aantal_aktief%>6
    Print At(5,24);"----> Meer dan ze =>"
    s accessoires aktief !! <----"
    Pause 100
    @Lijst
  Else
    For IX=1 To Aantal_accs%
      If Is_veranderd!(IX) Then
        Name "A:"+"Acsave0$(IX) As "A =>
: "+"Acsave1$(IX)
      Endif
    Next IX
    Edit
  Endif
  Return

Procedure Vulbox(Sx%,Sy%,Ex%,Ey%,Color%)
  Dx%=Abs(Sx%-Ex%)
  Dy%=Abs(Sy%-Ey%)
  L1%=Int(((Dx%*Dy%)/8)+2*Dy%)
  Vul$=String$(L1%+6,Chr$(255*Color%))
  Dpoke Varptr(Vul$)+0,Dx%
  Dpoke Varptr(Vul$)+2,Dy%
  Dpoke Varptr(Vul$)+4,1
  Put Sx%,Sy%,Vul$
  Return

Procedure Prohibit
  Cls
  If Resolutie%=2
    WZ=2
  Else
    WZ=1
  Endif
  Print String$(WZ*40,"*")
  L$=" "+Space$(WZ*40-2)+"*"
  Print At(1,2);String$(7,L$)
  Print At(1,8);String$(WZ*40,"*")
  If Resolutie%=2
    Print At(30,2);"A C C - L O A D E =>"
    R";Space$(25);"V1.3"
    Print At(15,4);"Je hebt twee seco =>"
    nden om AccLoader te stoppen : "
    Print At(13,5);Neg$;"!";Space$(24 =>
);"!";Space$(24);"!"
    Print At(15,6);Pos$;"<return> = s =>"
    top
    <spatie> = start meteen"
    Print At(13,5);"!";
  Else
    Print At(11,2);"A C C - L O A D E =>"
    R";Space$(5);"V1.3"
    Print At(5,4);"2 sec. om AccLoade =>"
    r te stoppen : "
    Print At(7,5);Neg$;"!";Space$(12) =>
;"!";Space$(11);"!"
    Print At(3,6);Pos$;"<ret>=stop =>"
    <spatie>=start meteen"
    Print At(7,5);"!";
  Endif
  Return

For IX=0 To WZ*25
  Print " ";
  Key$=Inkey$

```

```

  If Key$<>" "
    If Key$=" "
      IX=51
    Else
      Edit
    Endif
  Endif
  If IX=0 Or IX=25 Or IX=50
    Print Chr$(8);"!";
  Endif
  Pause 4/WZ
  Next IX
  Print
  Return

Procedure Lees
  Buffer$=Space$(7*512)
  Void Bios(4,2,L:Varptr(Buffer$),7,1 =>
1,0)
  IX=0
  Repeat
    F$=Mid$(Buffer$,IX*32+1,12)
    Ext$=Mid$(F$,9,2)
    Atr%=Asc(Mid$(F$,12,1))
    Del$=Left$(F$,1)
    If Ext$="AC" And (Atr%=0 Or Atr%=>
&H20) And Del$<>Chr$(229)
      Acc$=""
      For JX=1 To 11
        If Mid$(F$,JX,1)<>" " Then
          Acc$=Acc$+Mid$(F$,JX,1)
        Endif
        If JX=8
          Acc$=Acc$+","
        Endif
      Next JX
      Inc Aantal_accs%
      Acc$(Aantal_accs%)=Space$(12)
      Rset Acc$(Aantal_accs%)=Acc$
      Accundo$(Aantal_accs%)=Acc$(Aan =>
tal_accs%)
      Acsave0$(Aantal_accs%)=Acc$
      Acsave1$(Aantal_accs%)=Acc$
    Endif
    Inc IX
  Until IX=>112 Or Left$(F$,1)=Chr$(0)
  If Aantal_accs%=0
    Sput Nohelp$
    Print At(19,21);Chr$(7);" Ge =>
en accessoires op deze schijf "
    Void Inp(2)
    Edit
  Else
    Sget Nohelp$
  Endif
  Return

Procedure Help(NX)
  Sget Nohelp$
  If Resolutie%=2
    SpX=33
  Else
    SpX=1
  Print At(1,1);Neg$;String$(40,"=")
  Print At(1,16);String$(40,"=")
  Print At(1,2);" AAA " =>
L
  Print At(1,3);" A A CCC CCC =>
L
  Print At(1,4);" AAAAA C C =>
L
  Print At(1,5);" A A C C =>
L
  Print At(1,6);" A A CCC CCC =>
LLLL
  Print Pos$
  Endif
  Print At(SpX,8);" Selecteer met Up =>
Arrow en DwnArrow;
  Print At(SpX,9);" Inaktiveer met pi =>
jltje -->
  Print At(SpX,10);" Aktiveer met p =>
ijltje -->
  Print At(SpX,11);" Reset met Undo, =>
Klaar met Return.
  Print At(SpX,12);" Help voor hulp . =>
..
  Print At(SpX,13);"
  Print At(SpX,14);" NB: Dit programma =>
a werkt niet goed als
  Print At(SpX,15);" GDOS al is ingel =>
aden (GEMVDI resident).";
  If Resolutie%=2
    Print " "
  Else
    Print
  Endif
  Void Inp(2)
  Sput Nohelp$
  Print At(1,NX);
  Return

```



# HOE ZIT DAT MET DIE FLOPPY

Zou u het nu wel eens precies willen weten? U koopt floppies voor een diskdrive van het type SF354 waarvan de reclame soms zegt dat er 360K op gaat, en soms dat er 500K ongeformatteerd op gaat. In werkelijkheid blijkt de capaciteit 357.376 bytes te zijn. Maar sommige formatterprogramma's beweren er toch weer meer op te krijgen. Tenslotte is er dan nog zoiets als de Amiga die op hetzelfde schijfje 440K kan zetten...

Boeken en tijdschriften over de Atari ST zijn er genoeg, maar wat je weten wil is vaak moeilijk te vinden. Een oplossing is het zelf uit te zoeken. Een eerste stap op de weg naar de waarheid is de vage aanduiding dat onze floppy 'IBM-compatibel' zou zijn. Laten we de feiten maar eens op een rijtje zetten.

## 500K ONGEFORMATTEERD

1K is, zoals in de computerwereld gebruikelijk, 1024 bytes. In dit geval blijkt dat niet te kloppen: er gaan echt maar 500.000 en geen 512.000 bytes op een kant van een schijf. Hierbij wordt uitgegaan van 80 sporen. Per spoor dus  $500.000/80$ , dat is 6.250 bytes. Helaas kun je een floppy niet zo maar vol schrijven. Er moet eerst wat structuur worden aangebracht. Die structuur wordt gevormd door de sectoren die het formatterprogramma op bepaalde plaatsen van de floppy schrijft. Van die sectoren zien we (als we er op de goede manier naar kijken) 512 data bytes. Maar er is meer informatie. Een deel van die informatie is niet-informatie, namelijk de 'gap'. De gap is een niet beschreven ruimte tussen twee sectoren. Waar is dat nu voor nodig?

In deze wereld is niets geheel volmaakt: dat geldt ook voor de floppy. Neem eens aan dat we een drive hebben die 5% te langzaam draait. Het formatteren gebeurt via de floppy-controller hardware, die door een kwartsklok gestuurd wordt. Die klok is bijna net zo goed als uw horloge: binnen onderdelen van promilles nauwkeurig. De tijd voor het schrijven van een sektor is daardoor de standaardtijd die de ontwerper van het systeem in gedachten had. Maar de sektor wordt wel 5% korter als de drive 5% te langzaam draait. Gaan we die net geformatteerde floppy nu beschrijven in een drive die precies de goede snelheid heeft dan zal 5% van de informatie buiten de sektor vallen. Gelukkig is die gap er: daardoor valt alles nog net binnen de grenzen. Het verhaal blijft natuurlijk hetzelfde als de ene drive 2.5% te langzaam draait en de andere 2.5% te snel. Nemen we van de eerder genoemde 6250 bytes 95% dan houden we 5937 over. Zorgen we ervoor nooit meer dan dat op een spoor te schrijven, dan kunnen we foutloos werken met drives met genoemde tolerantie. Bij het lezen zouden veel grotere toleranties mogelijk zijn, omdat de klok dan wordt afgeleid uit de gelezen informatie. Het probleem zit, zoals gezegd, in het schrijven.

Maar er staat, behalve databytes en gaps, nog meer op een spoor. Iedere sektor wordt voorafgegaan door speciale codes die het synchroniseren van de leescircuits mogelijk maken. Die circuits moeten namelijk de eigenlijke databits kunnen onderscheiden van de klokbits. Omdat er maar één spoor tegelijk wordt gelezen, is dat niet zo gemakkelijk. Ook wordt iedere sektor voorafgegaan door informatie-bytes die iets zeggen over de kop (de enige, of één van de twee bij een dubbelzijdige floppy), over de sektor en over het spoornummer, zodat we zeker zijn dat we met de juiste sektor aan het werk gaan. Daarnaast staat er op ieder spoor ook een indexmark, dat als het ware het nulpunt van een spoor vormt: van daaruit worden de sectoren genummerd. Tellen we alles bij elkaar en houden we rekening met die 5% variatie in snelheid dan blijken er bijna 9.5 sectoren op een spoor te gaan. Een halve sektor past op geen enkele manier in het systeem. Daarom gebruiken we 9 sectoren van 512 bytes per spoor:  $80 \times 9 \times 512 = 360K$ . Zowel 500K als 360K is dus juist, alleen is de eerste 'K' 1000 en de tweede 1024.

## SCHIJVEN OPVOEREN

Wie die 6250 bytes per spoor maar niet vergeten kan, zal zich afvragen of al die gaps niet wat kleiner kunnen. Dat blijkt inderdaad zo te zijn: de meeste drives zijn veel beter dan Atari denkt. De moderne motorstuurlektronika heeft er niet veel moeite mee

om de snelheid binnen 1% konstant te houden. Dan mag je volgens het bovenstaande verhaal 99% van 6250 gebruiken, dat wil zeggen 6187 bytes. Daar blijken 10 sectoren net lekker in te passen. Dat is een sektor extra per spoor, of 80 per floppy, oftewel 40K bytes. Dat blijkt de truuk te zijn die sommige formatterprogramma's gebruiken.

Is de garantietermijn van de drive al bijna voorbij? Kijk dan eens voorzichtig in de diskdrive. Er blijkt 'direkt-drive' toegepast te zijn: de motor drijft direkt de floppy aan. Hoe hard gaat dat eigenlijk? Dat is vast te stellen met de functie "omw/minuut" uit het bijgevoegde GfA-programma.

Mijn SF354 blijkt 299 toeren te maken: de nominale waarde is 300. Mijn drive draait dus  $1/3\%$  te langzaam: ruim binnen de marge van 1%. De 'bios' aanroep zorgt ervoor dat steeds sektor S van de floppy gelezen wordt (functie 'rwabs'). Wie die AS niet begrijpt: hierop kom ik verderop in dit artikel terug.

Het buitenste spoor (spoor 0) heeft een doorsnede van ongeveer 8 cm. De lengte is dus zowat 25 cm. Daarop passen de eerder genoemde 6250 bytes. Een bit op dat spoor heeft een 'lengte' van  $250/(6250 \times 8) = 0.005$  mm. Niet zo veel, maar op het binnenste spoor is een bit nog veel kleiner. De leeskop heeft daar moeite mee. In feite bepaalt de maximale dichtheid van het binnenste spoor de capaciteit van de floppy. Dat binnenste spoor (spoor 79) is niet het uiterste wat het mechaniek dat de kop verplaatst kan bereiken. Dat mechaniek (bestuurd door een stappenmotortje + elektronika) heeft geen echte begrenzing. Als je een programma maakt dat bewust probeert 'te ver' te gaan, hoor je een angstwekkend geraas van het kopmechanisme dat tot aan de as van de floppy probeert te komen. Dat kan natuurlijk niet, maar het is wel zo dat de sporen 80, 81 en 82 bereikbaar en bruikbaar zijn. Daarvoor moet de kop in prima staat zijn. Eventuele slijtage betekent dat er zich bij de sporen boven de 79 het eerst problemen voordoen. Niemand zegt dat je die sporen echt mag gebruiken. In de praktijk blijkt dat echter uitstekend te gaan. Dat is dan ook de tweede truuk die de formatterprogramma's gebruiken. Een heel goed voorbeeld staat in START 2, nov/dec 1986.

Standaard is dus  $80 \times 9 = 720$  sectoren. Truuk één levert 80 sectoren op en truuk twee nog eens  $3 \times 10$ . Samen 110, zo'n 15% winst.

Truuk nummer drie is helaas niet uitvoerbaar, maar dit is hem: als je ontdekt dat de maximale dichtheid nog wel wat vergroot mag worden, kan je ook de klokfrequentie opvoeren. Dan moet het lukken om 11 sectoren op een spoor te krijgen. Om het niet te gek te maken moet je dan niet meer dan 80 sporen gebruiken. Dat zou dan  $80 \times 11 \times 512 = 440K$  op leveren. Het zou wel eens kunnen dat de Amiga zoiets doet, maar details zijn mij niet bekend. (Die kunnen ons als Atari-fans ook niets schelen). Het lijkt mij geen prettige ingreep om die klokfrequentie van de floppy controller zo'n 10% op te voeren.

## ADMINISTRATIE

Als je een floppy formatteert met 10 sectoren en 83 sporen, blijkt dat TOS daar weinig problemen mee heeft. Bestanden kopiëren, folders openen, enzovoorts gaat allemaal feilloos. Alleen een 'standaard' floppy overschrijven naar eentje met 830 sectoren door in de desktop twee floppy iconen op elkaar te leggen werkt niet: dat gaat alleen met floppies die precies hetzelfde formaat hebben. De vraag is nu, hoe weet TOS met wat voor floppy er gewerkt wordt? Het antwoord is: door in de 'bootsektor' te kijken.

Sektor 0, spoor 0, kant 0 is de zogenaamde bootsektor. In die sektor kan een zogenaamde 'bootstrap-loader' staan. Dat is het geval bij de 'TOS-floppy'. De meeste andere floppies hebben die loader niet. Wel hebben ze allemaal een hoeveelheid gegevens. De belangrijkste zijn:

byte(s)	waarde	betekenis
8-10	?????	serienummer
11-12	512	bits per sektor
13	2	sektors per cluster
14-15	1	gereserveerde sektor (sektor 0 in dit geval)
16	2	aantal FAT's. Wat dat zijn komt nog.
17-18	112	maximum aantal regels in de directory
19-20	720	aantal sectoren. Dat kan dus ook 800 tot 830 zijn, en bij een dubbelzijdige floppy 2x zoveel.
22-23	5	aantal sectoren per FAT
24-25	9	aantal sectoren per spoor. Dat kan dus ook 10 zijn.
26-27	1	aantal kanten: 2 kan ook.

Het serienummer is bij de 'standaard' formattering een toevalsgetal van 24 bits. TOS gebruikt het om schijfwisseling vast te stellen. Sommige formatterprogramma's uit het 'Public Do-



main' vinden het makkelijker om daar een nul te schrijven. Daardoor zou het wel eens kunnen dat TOS het niet merkt als er een schijf gewisseld wordt. Het programma uit START doet dat zoals het hoort.

TOS bekijkt de floppy in eenheden van 1024 bytes: dat is de zogenaamde 'cluster' die dus uit twee sectoren bestaat. Je zou je ook sectoren van 1K kunnen voorstellen. Het begrip 'cluster' zou dan overbodig zijn. Ga je er echter van uit dat er maar 9x512 bytes op een spoor gaan, dan hou je een halve cluster over: een aanzienlijke verspilling. Door clusters te gebruiken hoeft je maar van half zoveel dingen te onthouden hoe hun toestand is: de administratie wordt minder omvangrijk. Anderzijds neemt nu het allerkleinste bestandje al 1K diskruimte in beslag. Die ene K is dus een kompromis: met clusters van 10K is er nog veel minder te administreren, maar is de 'overhead' bij kleine bestanden nog groter.

De eigenlijke administratie vindt plaats in de FAT, de File Allocation Table. Daarin wordt genoteerd welke clusters vrij en welke bezet zijn, en hoe de samenhang tussen meerdere clusters in een bestand is. Omdat de FAT heel belangrijk is, staat hij twee keer op de disk. Daar gaan we straks nog wat dieper op in.

Met 'directory' wordt hier de eerste directory bedoeld. (Atari-fans spreken meestal van een 'folder'). Een regel in de directory bevat meer dan alleen de bestandsnaam. Daardoor zijn er 32 bytes nodig. Er passen 112 van dergelijke regels in de directory. Hieruit blijkt dat hij  $32 \times 112 / 512 = 7$  sectoren beslaat. Ook hier komen we nog op terug.

Wie in staat is zijn disk te lezen zal de genoemde informatie niet herkennen. Dat komt omdat alle byte-paren verwisseld zijn: de 'meest-belangrijke' staat rechts. Dit is om historische redenen: omdat je bij optellen en aftrekken rechts moet beginnen, vonden sommige computerfabrikanten het handig om getallen achterstevoren in het geheugen te zetten. Daar het ook wel handig is als een nieuw produkt met een ouder kan samenwerken, blijft die oude methode nog steeds in gebruik. In de sectoren die we gewoonlijk lezen en schrijven staan de gegevens wel in de 'gewone' volgorde, dus net zo als in het geheugen.

## DE FILE ALLOCATION TABLE

Sektor 0 is, zoals we nu weten de bootsektor. De eerste FAT staat in de sectoren 1 t/m 5; de kopie ervan in de sectoren 6 t/m 10.

De FAT bevat een lijst met gegevens over clusters. Ieder gegeven is een, schrik niet, 12 bits getal. In een sektor passen  $512 \times 8 / 12 = 341.3$  van die gegevens. In totaal dus  $5 \times 341.3 = 1706$ . Wie kans ziet om 1.7M op z'n floppy te krijgen zit nog steeds goed. De maat van de FAT is alweer het gevolg van de gewenste compatibiliteit.

Die 12 bits zitten nog meer door elkaar gehaspeld dan in de bootsektor. In het volgende tabelletje stellen de letters telkens 4 bits voor:

A B C D E F G H I J K L    3X16=48 bits in de floppy-FAT  
D A B E F C J G H K L I    4 getallen van 12 bits

Niet echt gemakkelijk te lezen. Wie dit even tekent, ziet dat het voldoende is om groepen van 24 bits te bekijken: drie bytes leveren op steeds dezelfde manier twee cluster-nummers op. De positie van het 12-bits nummer hangt samen met de cluster waarop het betrekking heeft. Het tiende nummer zegt dus iets over cluster 10. Het 12-bits nummer kan de volgende waarden hebben:

- 0 een vrije cluster
- 1 de laatste cluster van een bestand (soms de enige)
- 9 niet bruikbare cluster
- >0 een verwijzing naar de volgende cluster.

Op positie 23 kan dus een verwijzing naar 24 staan, daar weer naar 25, dan naar 26 en daar zou -1 kunnen staan. Het bestand bestaat dan uit 4 clusters, de lengte van het bestand is iets tussen de 3073 en 4096 bytes. De clusters staan niet altijd zo netjes op volgorde. Op een floppy waar wat bestanden gewist zijn komen 'gaten' voor. Bij het schrijven van een relatief groot bestand worden deze gaten opgezocht en na elkaar beschreven. De clusters van dat bestand zullen dan waarschijnlijk niet opeenvolgend zijn. Wel zal een volgende cluster altijd een hoger nummer hebben dan een vorige. Bekijk het allemaal zelf maar eens met de funkties 'hexa' of 'ASCII' onder menuhoofd 'sector' uit het GfA-BASIC programma.

## DE DIRECTORY

De meeste Atari-gebruikers bekijken folders in 'ikoonvorm'. Ik zie ze liever als tekst, omdat dan zowel de tijd van de laatste wijziging als de lengte van een bestand zichtbaar zijn. Op de schijf blijkt echter nog iets meer te staan.

Een 'regel' in de directory begint met de bestandsnaam: 11 bytes zonder de scheidingspunt. Als het eerste byte van de naam hexadecimaal 'e5' is, dan is dat bestand gewist. 'Gewist' betekent niet dat alle informatie gewist is, maar wel dat alle clusters van de file vrijgegeven zijn.

Na het formatteren zijn de sporen 2 tot 79 (of 82) volgeschreven met diezelfde 'e5', tenzij de formateur in zijn wijsheid besloot een ander vulpatroon te gebruiken. Dat mag, maar 'e5' is min of meer standaard. Hierdoor is een nooit gebruikte sektor herkenbaar. (Gesteld dat je die zou willen herkennen.)

Om terug te komen op de directory: met het bijgeleverde GfA-BASIC programma kunnen we sektor 11 eens bekijken. (Even repeteren: een bootsektor en twee FAT's van 5 sectoren, dat zijn er 11, nummers beginnen bij 0, de twaalfde sektor is nr. 11!)

Wie goed is in ASCII zal aan het begin van iedere regel een bestandsnaam herkennen. Die wordt gevolgd door een byte dat het 'attribute' van het bestand bevat. Meestal, maar niet altijd, is dat een nul-byte. De volgende tabel toont alle mogelijkheden:

byte (hex!)	betekenis
00	een 'normaal' bestand
01	read only
02	hidden (wordt niet op desktop getoond)
04	system
08	volume label (floppy-naam die bij het formatteren is opgegeven)
10	subdirectory (folder)

Hogere waarden worden voor de floppy niet gebruikt. Wel zijn sommige combinaties van bits denkbaar.

Via een desktop funktie kan een bestand 'read only' gemaakt worden. Het 'attribute' blijkt dan inderdaad de waarde 01 te krijgen. Via de desktop kunnen ook nieuwe folders gemaakt worden. Die zijn nu eveneens gemakkelijk te herkennen.

## EXTRA INFORMATIE

Na naam en attribute blijken 10 gereserveerde bytes te volgen. Door het BASIC-programma worden die als vijf nul-woorden getoond. Daarop volgen woorden die respectievelijk tijd en datum van de laatste wijziging voorstellen. Om die te ontcijferen is het allereerst nodig om linker- en rechterbyte van ieder woord te verwisselen. Dan staat er in 5-6-5 bits: uur-minuut-sekunde. De ervaren binaire rekenaar zal opmerken dat 60 seconden niet in 5 bits passen. Daarom worden seconden per twee geteld. Anders zou de tijd niet in het handzame formaat van 16 bits passen. De datum is genoteerd als 7-4-5 bits: jaar-maand-dag. Bij het verkregen jaarnummer moet nog 1980 worden opgeteld. In 7 bits past maximaal 127. U zult dus in het jaar 2107 over een andere computer moeten gaan denken.

Het volgende woord is interessant: het bevat iets wat de desktop niet toont, namelijk het nummer van de eerste cluster van een bestand. Ook hier staan linker- en rechterbyte verwisseld. Het zou moeten lukken om dat bestand te vinden, als we wisten hoe clusternummers en sektornummers samenhangen. Dat blijkt redelijk logisch, maar toch niet helemaal vanzelfsprekend te zijn. De ontwerper van de FAT wist dat een floppy altijd begint met bootsektor, FAT, nog een FAT, eerste directory. Dat zijn  $1+5+5+7=18$  sectoren. De eerste echt vrije sektor is de 19de. Nummer 18 dus. De eerste vrije cluster bestaat uit de sectoren 18 en 19. In de bespreking van de FAT bleek dat een cluster-nummer 0 betekent dat de cluster vrij is. Een verwijzing naar cluster 0 is dus niet mogelijk. Daarom begint de clusternummering bij 1. Bent u daar nog? Cluster nummer 1 blijkt ook niet gebruikt te worden, daarom verwijst cluster nummer 2 naar de sectoren 18 en 19, 3 naar 20 en 21, en zo verder.

Achttien sectoren zijn dus vast in gebruik. Er zouden er dus  $720-18=702$  voor de gebruiker beschikbaar moeten zijn.  $702 \times 512 = 359.424$ , dat is 2048 meer dan 357.376. Dat laatste getal wordt door de desktop opgegeven na het formatteren van een enkelzijdige floppy. Waar zijn die 2048 bytes nu gebleven? Het volgende experiment laat dat zien.

Formateer een enkelzijdige floppy. Kopieer bestanden totdat de floppy precies vol is. (De laatste bestanden moeten < 1K zijn.) Bekijk de floppy met het GfA-programma. Dan blijken de sectoren 716-719 nog steeds 'e5' te bevatten: ze zijn nog ongebruikt. Nu klopt het cijfer wel, maar waarom worden die laatste twee clusters niet gebruikt? Ik heb geen idee..! Ter nadere controle: bekijk de directory van die helemaal volle floppy. Deel de lengte van ieder bestand door 1024, rond het antwoord af naar boven. Tel alle antwoorden bij elkaar. De som zal 349 zijn.  $349 \times 1024 = 357.376$ , dat klopt dus.

Let op: dit is een verhaal over floppies. Het is niet zondermeer waar voor hard disks!



Na de deklaratie van A\$ wordt er een menu gestart. Via het menu

Ook is kritiek mogelijk op de weergave van sectoren als hexadecimale woorden of als ASCII bytes. Dat kan met wat passen en meten samen op een scherm. Dan wordt het ook interessant om dat scherm 'editbaar' te maken. Voer voor krakers, zou ik zeggen.

```

Procedure Drline(X)
  Draw 0,32+X To 639,32+X
Return

```



# OFFICIAL ATARI ST-DEALER

## ZAKELIJKE SOFTWARE

### DATABASES

Astodat II	f 299,-
Laserbase	f 298,-
ST Datamanager	f 299,-
Habaview	f 149,-
Hippo Simple	f 99,-
DB Man	f 359,-
Trim base	f 398,-

### FINANCIAL SOFTWARE

ST Financial Junior	f 149,-
ST Financial Personal	f 449,-
ST Financial Professional	f 719,-
ST Financial Voorraad	f 395,-
Projecta-4 ST	f 829,-
Divi Boek	f 198,-
Makro Boek	f 369,-

### DIVERSEN

K-switch	f 109,-
K-graph	f 99,-
K-graph II NIEUWII	f 198,-
K-spread	f 249,-
Vip 1.2.3 (GEM)	f 798,-
Swift Calc	f 299,-
Easy Calc	f 169,-
Habadex phonebook	f 119,-
Haba-spell	f 79,-
Haba mailmerge	f 79,-
GST 1st Mail	f 99,-

### TEKSTVERWERKING

Habawriter II	f 199,-
Signum	f 549,-
Color Writer	f 99,-
Wordstar 3.0 (cpm)	f 198,-
Textomat	f 127,50
Final Word	f 398,-
Regent word/spell	f 198,-
K-word	f 179,-
Word Writer	f 299,-
Thunder	f 99,-

### MODEM SOFTWARE

Viewdata st	f 298,-
Gem viditel	f 298,-
VT 100 Terminal	f 99,-
PC intercomm	f 249,-
K-comm v2.0	f 179,-
Mi-term	f 89,-
Flash	f 139,-
Dos shell	f 99,-
Real time clock	f 179,-
Z-Time (internal clock/call.)	
voor inbouw	f 199,-
Micro time (int. clockcard.)	f 199,-

### PROGRAMMEERTALEN

G.F.A. basic v2.0	f 215,-
GFA Basic Com.	f 210,-
Fast basic	f 399,-
Megamax c	f 725,-
G.F.A boek + disk	f 99,-
Metacomaco pascal	f 349,-
Mark Williams C	f 599,-
K-seka assembl.	f 79,-
Devpac hisoft	f 189,-
Pro-Fortran	f 549,-
Cambridge lisp	f 549,-
Expert Opinion	f 325,-
Lattice C compiler metacomaco	f 398,-
G.S.T.C compiler	f 199,-
L.D.W basic	f 399,-
Modula 2	f 449,-
ST Pascal	f 298,-
U.C.S.D. pascal	f 498,-
Macro assembler	f 199,-
GST Macro assembler	f 149,-
B.C.P.L	f 399,-
Profimat ST	f 125,-

### PROGRAMMEER UTILITIES

K-Resource	
(resource constructie set)	f 149,-
Menu +	f 89,-
Modula 2 toolkit	f 199,-

### UTILITIES ALGEMEEN

ST toolbox	f 139,-
Disk help	f 139,-
M-disk (ram d)	f 149,-
Kissed (debug.)	f 89,-
Toolbox robtek	f 119,-
Crystal	f 89,-
Lock Smit	f 169,-
ST Mult. hardcopy	f 155,-
Michtron utilities	f 119,-
Hippo disk utilities	f 199,-
ST Key	f 99,-
Super Directory	f 99,-

### DESCACC

Cornerman	f 79,-
Rhythm	f 49,-
Powerpak diary	f 79,-
Desk assist	f 199,-
Macro manager	f 149,-
Powerpak database	f 79,-
Powerpak adressbook	f 49,-

### TEKENPROGRAMMA'S

STAD	f 239,-
Degas	f 99,-
Profi painter	f 125,-
Picop (tek. ut.)	f 49,-
G.S.T frac gen	f 69,-
Printmaster	f 129,-
Colour space	f 59,-
Monostar	f 129,-
Monostar plus	f 179,-
Degas Elite	f 189,-
Datamap col I	f 69,-
Gfa Draft (cad)	f 369,-
Art Director	f 179,-
Turbo dizer	f 798,-
Publishing Partner	f 499,-
Fleetstreet publisher	f 459,-
Type Setter Elite	f 119,-
Text design st	f 127,50
Paintworks	f 149,-
Easy draw	f 298,-
Animator michtron	f 119,-
Mica	f 249,-
Platina ST	f 490,-
Cad-3D	f 149,-
Future Design Disk	f 79,-
N-Vision	f 159,-
Maps & Legends	f 129,-
Colorstar	f 129,-
Gfa Vektor	f 189,-
Film Director	f 199,-
Sprite constr. set	f 79,-

### MUZIEK PROGRAMMATUUR

Music studio	f 139,-
Midisplay	f 99,-
Steinberg research	
24 sporen recorder	f 539,-
Steinberg research Voodoo	
(random sound generator)	f 439,-
Treesoft midi-recorder	f 250,-
Musidisc (midiplay lib.)	f 59,-
G.I.S.T (sound tool)	f 199,-
Digidrum	f 159,-

### ADVENTURES

Infocom adventures uit voorraad leverbaar	
Oper. Hong Kong	f 89,-
Transylvania	f 99,-
Fahrenheit	f 119,-
Treasure Island	f 119,-
Crimson Crown	f 99,-
Perry Mason	f 119,-
The pawn	f 99,-
Hacker	f 99,-
Hacker II	f 119,-
Exodus Ultima III	f 99,-

Mind Shadow	f 119,-
Borrowed time	f 119,-
Ultima II	f 119,-
Spiderman	f 49,-

### GAMES

World Games	f 79,95
Leader board	f 99,-
Arena	f 79,-
Psion Chess	f 99,-
Tee-Up golf	f 59,-
High Roller	f 129,-
Wintergames	f 99,-
Flipside	f 49,-
Mean 18 golf	f 119,-
Match Point	f 99,-
Shuffle board	f 79,-
Karate master	f 49,90
Major Motion	f 89,-
Lands of Havoc	f 99,-
Xtron	f 99,-
Red alert	f 49,-
Little comp p.	f 99,-
Deep space	f 129,-
Gato	f 109,-
Silent service	f 99,-
Space Station	f 79,-
ST Karate	f 95,-
ST Warzone	f 59,-
Starglider	f 99,-
Mercenary Comp.	f 99,-
Q-Ball	f 69,-
Electr. Pool	f 79,-
Flicht Simul. II	
(Nul voorkl/mon)	f 179,-
Thai Boxing	f 49,50
Karate Kid II	f 85,-
Balance of Power	f 139,-
Phantasie II	f 129,-
Shuttle II	f 79,-
Liberator	f 59,50
M.G.T	f 99,-
Metrocross	f 99,-
Road War 2000	f 129,-
Brataccas	f 79,-
Mudpies	f 59,-
Monkey business	f 79,-
Black cauldron	f 149,-
Rogue	f 119,-
ST Protector	f 79,-
ST Fireblaster	f 59,-
Temple of Apshai	f 109,-
Strip Poker	f 69,-
Terrestrial Encounter	f 59,-
Timeblast	f 39,95
Two on Two basketball	f 99,-
Space Pilot	f 49,50
Championship Wrestling	f 69,95
International Karate	f 79,-
Typhoon	f 99,-
Joust	f 59,-
Plutos	f 59,50
Wanderer	f 95,-
Goldrunner	f 99,-
Mouse Trap	f 99,-

## HARDWARE

### HARDWARE

Atari 1040 stf/sm 125	f 2399,-
Atari 1040 stf/sc 1224	f 2998,-
Atari 520 stM/sf 354	f 1198,-
Modulator voor Atari ST	f 269,-
Nu verkrijgbaar!! Kleuren Digitizer	
voor 260/1040 ST	f 498,-
Ook!!! Mouse Matjes v.a.	f 19,95
Compactkast ST	f 169,-

Leveringen geschieden onder rembours door heel Nederland. De rembourskosten variëren van 12,50 voor programmatuur tot 25,00 voor hardware. (eventuele prijswijzigingen voorbehouden.)

OFFICIAL ATARI REPAIR CENTER

Prijswijzigingen voorbehouden. Alle prijzen incl. BTW

# TELEKODER

## COMPUTERS

Nederlands grootste computer speciaalzaak - Verzending door heel Nederland

ATARI SPECIAL SHOP TEL. 010-4148605

Hoogstraat 26-53-65 Rotterdam. Postorders tel. 010-4133495



Dat START alles wat met tekenen te maken heeft een warm hart toedraagt, merkt u elk nummer aan de voorplaat van Robert van der Kroft.

Die samenwerking leidde er toe dat Robert in nummer twee het animatieprogramma FILM-DIRECTOR testte. Er is nu een concurrent: importeur SOFT-ST stuurde ons een recensie exemplaar van ANIMATOR. Uiteraard vroegen we Robert om ook ditmaal als expert op te treden.

## LEKKERST VOOR HET LAATST

ANIMATOR, van het Amerikaanse softwarebedrijf Aegis, is ontwikkeld voor de Commodore Amiga en pas later omgeschreven voor de ST. Kennelijk groeit de ST-markt in Amerika momenteel als kool en was het niet langer verantwoord om die markt te laten liggen. Een goed teken. Dat omschrijven is helaas niet vlekkeloos gebeurd: de software heeft (nog?) wat mankementen. Laten we gelijk maar even door de zure appel heen bijten en de bugs ter sprake brengen. Het lekkerst bewaren we zoals het hoort voor het laatst. Blijven lezen dus!

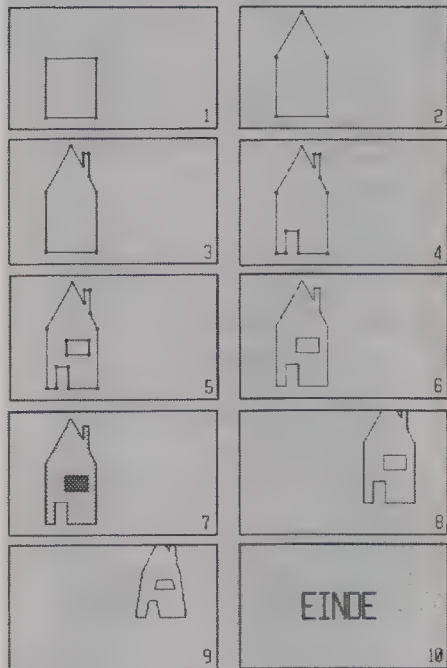
Allereerst is het heel vervelend dat het programma slecht naar de muisklik luistert. Het gebeurt keer op keer dat je denkt dat je een nieuwe functie activeert en dan later merkt dat de vorige functie nog "aan" staat. Een tweede bug bevindt zich in het kleurenmenu. Als je de cursor over de rode kleur zet, wordt hij blauw. Zet je hem over blauw, dan krijg je rood. De kleur die het blokje heeft voordat de muis er op staat is echter bepalend. Kortom: eerst kiezen en dan pas de muis bewegen. Tenslotte lukte het me enkele keren de menubalk in de war te maken. Na een beetje rommelen werd alles echter weer normaal. Hinderlijke bugs, maar gelukkig geen fatale.

## GROEIENDE SCHOORSTEEN

Bent u daar nog? We hebben de kritiek gehad. Voor de rest is Animator gewoon een heel erg leuk programma om mee te spelen.

Het principe van het pakket is gebaseerd op veelhoeken. Elke willekeurige figuur is op te bouwen uit veelhoeken. Laten we eens een tekenfilm maken getiteld "het huisje dat kon vliegen". We tekenen daarvoor een vierkant op het scherm (fig. 1). Daarna maken we midden in de bovenste ribbe van het vierkant een nieuw hoekpunt en trekken dat omhoog om een dak te maken (fig. 2). Op dit moment hebben we al een filmpje! Animator berekent namelijk alle beeldjes die tussen figuur 1 en 2 liggen. Bij afspelen zien we het dak dan ook langzaam groeien. Toch is het eigenlijk niet helemaal juist om van "tussenvormende beeldjes" te spreken. Als we wijzigingen willen aanbrengen kan dat namelijk alleen in figuur 1 en figuur 2. De overgangsfase houdt Animator voor zichzelf. Op dezelfde manier kunnen we vier punten aan de rechter zijkant van het dak toevoegen en een schoorsteen laten groeien (figuur 3). Een deur (figuur 4) is van hetzelfde laken een pak. Tot nu toe zijn we er in geslaagd

# ANIMATOR: De kunst van het vervormen



tekeningen DEGAS

het huisje uit slechts een veelhoek te laten bestaan. Als we ook nog een raam willen hebben (figuur 5), moet een tweede veelhoek toegevoegd worden.

Ons huisje ziet er nu uit als in figuur 6 of 7. Bij het afdraaien van het filmpje zijn nooit punten zichtbaar. Animator werkt naar keuze met contouren of massief.

Het wordt pas leuk als we het huisje laten vliegen. Je kunt het hele object beetpakken en naar een nieuwe plaats brengen. Zoals steeds: Animator berekent de overgang. Er is maar één ding waar we op moeten letten: dat het raam niet achterblijft. In het pick-menu bepalen we eerst wat we manipuleren: punt, punten, veelhoek, veelhoeken of alles. Veelhoeken of alles is in dit geval een juiste keus. Huis en raam worden als één object behandeld en verschuiven bij afdraaien gelijktijdig naar hun nieuwe plaats (figuur 8).

We kunnen het huisje ook roteren, vergroten, verkleinen of in een as spiegelen. Vooral dat laatste is erg indrukwekkend. Animator verandert het huisje voortdurend van perspectief, zodat er diepte in de tekenfilm komt (figuur 9).

Bovendien zijn er ook combinaties mogelijk: het huisje gaat dan van positie A naar positie B TERWIJL het bijvoorbeeld een draai van 180 graden maakt. Bijzonder imposant.

Natuurlijk kan het hele animatiefilmpje verrijkt worden met een zon die op en weer onder gaat en een voorbij schuivende wolk.

Zo'n wolk is snel gemaakt met een aantal cirkels. Animator heeft standaard cirkels en sterren in huis. Eigenlijk zijn dit natuurlijk ook gewoon veelhoeken, zij het dat zo'n cirkel nogal wat hoekpunten heeft.

## IMPORT

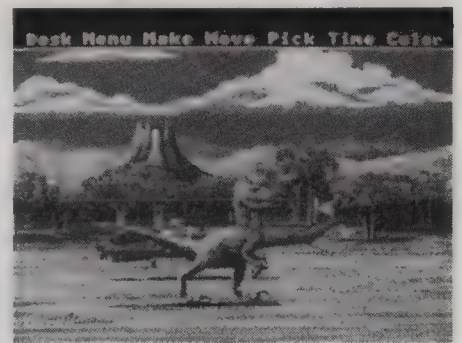
Kan Animator buiten veelhoeken ook gedetailleerde tekeningen animeren? Het antwoord is "ja maar...". Het is mogelijk een tekening in het Neochrome-formaat in te laden, daar stukken uit te snijden en die te verschuiven. Dit biedt dus ongeveer dezelfde mogelijkheden als Film-director, maar ik zal Animator niet snel op die manier gebruiken. Film-director laadt twee tekeningen in en onthoudt op welke manier ze in stukken zijn gesneden. Animator bewaart de stukken afzonderlijk op schijf. Dat betekent hfdlinks.cel, hfdrecht.cel, lijfvoor.cel, lijfzij.cel, enz. op schijf. De uiteindelijke film "weet" welke stukken er nodig zijn, maar tijdens het animeren zit je voortdurend met die disk-drive in je maag. Bovendien is er geen enkele manier om in Animator een tekening bij te werken. Daar moet telkens Neochrome weer voor worden geladen.

Wat mij betreft is de kompatibiliteit met Neochrome alleen interessant voor het toevoegen van decors.

## TEKST EN UITLEG

Animator beschikt over een heuse montagetafel. Zes stukken film kunnen gelijktijdig bewerkt worden. Dit is de plek om te knippen, te plakken en te kopiëren. Als leuk extraatje kan Animator de zes stukken "tegelijk" afspelen.

Een belangrijk onderdeel van een tekenfilm is de "timing". Bij Animator is die per handeling instelbaar. Als we vinden dat onze schoorsteen te langzaam groeit, geven we dat gedeelte van de film via het time-menu eenvoudig een kortere tijdsduur mee.



Het editen van een film kan echter ook met een tekstverwerker! Het programma slaat een film op als een ASCII-file. In principe is een tekenfilm daardoor ook helemaal in First Word te maken. De uitstekende Engelse handleiding zet de liefhebber daarbij prima op het spoor.

Animator is een prima pakket voor wie grotere vormen wil animeren. Van een titelrol voor een videoband met bewegende letters kan echt iets heel bijzonders worden gemaakt.

Animeren met Animator kost relatief weinig tijd. Voor een tekenfilm is eigenlijk niet meer dan een storyboard met key-drawings nodig die ook nog eens eenvoudig uit elkaar af te leiden zijn. Alle moeilijke tussenvormen berekent het programma. Op die manier maken vierentwintig beelden al een film van een seconde of tien. De ouderwetse manier van cel-animatie vult er net een seconde mee.

Wie een film klaar heeft, kan hem aan iedere ST bezitter kado doen. PLAYER, de bijgeleverde uitgeklede versie van Animator, kan alleen tekenfilms afspelen en mag vrij worden gekopieerd.



# ASTRO CHALLENGE

*De naam van de planeet is ASTRO. Het jaar: 2001. Een delegatie van geleerden heeft zojuist de beslissing genomen een onderlinge vete eindelijk te beslechten. De Astroïden zijn sinds hun bestaan in twee kampen verdeeld. De ene helft van de bevolking meent dat de planeet ASTRO plat is. De andere helft houdt staande dat hun wereld ellipsvormig is. Een expeditie zal de waarheid aan het licht moeten brengen.*

Kapitein Spot heeft de leiding over de luchtvloot die het onderzoek zal uitvoeren. Opdracht is om tot de rand van de planeet te vliegen of, in het geval de planeet ellipsvormig is, een volle baan om Astro te voltooien.

De luchtvloot, honderden schepen groot, vertrekt voorspoedig. Spot, zelf een overtuigd ellips-aanhanger, bestuurt het voorste vliegtuig, een G-17. Energiek doorbreekt hij de geluidsbarrière; MACH-3 volgt luttele seconden later.

Spot geniet groot aanzien bij zijn manschappen. In blind vertrouwen laten

daarom ook de Astroïden in de volgvliegtuigen hun snelheidsmeters oplopen.

Plots klinkt een schreeuw in de koptelefoons: "Kapitein! De planeet! Hij verdwijnt uit zicht!"

Een ongepaste grap van een van de luitnants. "Volle kracht vooruit!" beveelt Spot geërgerd en voegt meteen de daad bij het woord. De Astroïden in de volgvliegtuigen gehoorzamen slaats.

Pas dan richt Spot zijn blik naar beneden. Hij slaakt een kreet van afgrijzen. Astro is plat! Diep onder hem is een duidelijke rand te zien.

Helaas beperkt de grens zich niet alleen tot de planeet. De atmosfeer boven Astro is ook afgebakend. Met doffe klappen slaan de vliegtuigen tegen de rand.

Het einde komt snel en pijnloos voor Spot en zijn manschappen. Niemand overleeft de plotselinge verandering in snelheid.

Het leed voor de bewoners van de planeet vangt echter nu pas aan. De wanden van de atmosfeer hebben namelijk een vreemde uitwerking op de vliegtuigen. Ze stuiteren terug naar de bewoonde wereld en slaan gaten in de planeet. Woonkernen worden niet ontzien. Een enorme paniek breekt uit.

Het is aan u om Astro te hulp te komen en de spookschepen te vernietigen. U beschikt over drie verrijdbare raketbasissen, die u bestuurt met de muis.

Astro Challenge werd geschreven door Iban van der Wal voor gebruik op een monochrome monitor.

Misschien heeft u ooit op een 8-bitter een grafisch spel in BASIC proberen te schrij-

ven. Goede kans dat u tien kniebuigingen kon maken voor een sprite van de ene naar de andere kant van het beeldscherm was verhuisd. Astro Challenge demonstreert dat dit met GfA-BASIC en de ST verleden tijd is. U zult de ruimteschepen nauwelijks bij kunnen houden. En dan hebben we nog niet eens de compiler gebruikt.

Alleen het toevoegen van geluid was in eerste instantie een probleem. Op het moment dat het schietgeluid geproduceerd werd, stonden de sprites even stil. Die vertraging bedierf het visuele gevoel.

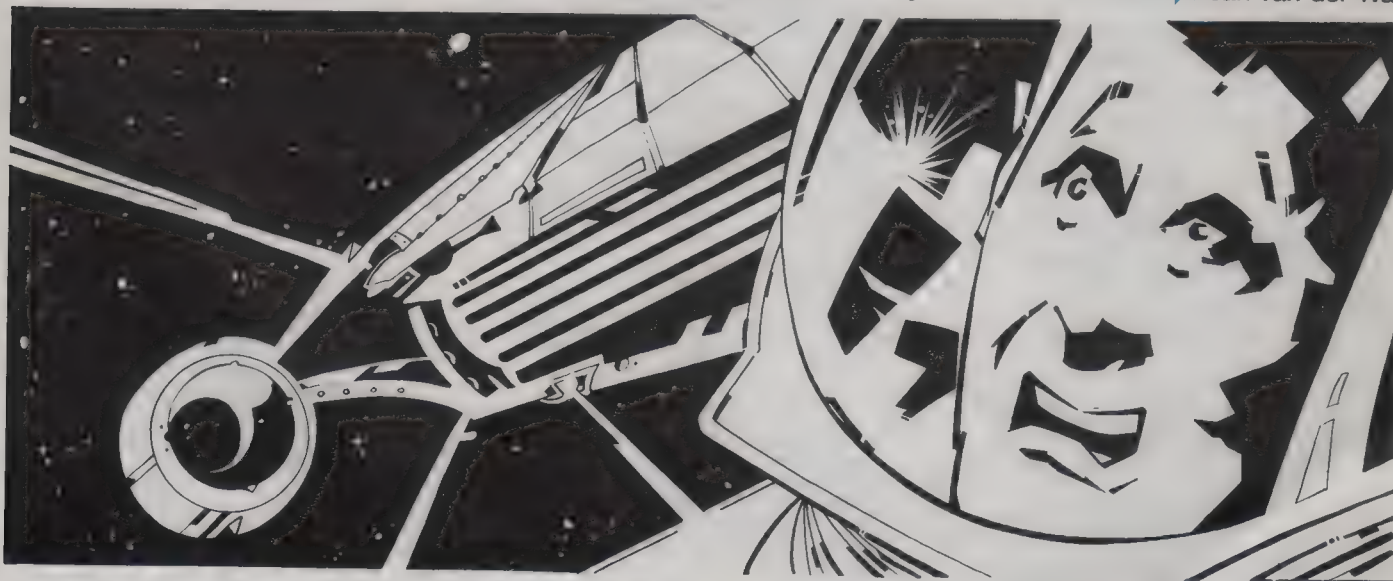
Gelukkig presenteerde Egon Lubbers in het vorige nummer van START een manier om geluid via een interruptroutine af te spelen. Een interruptroutine kan 'tegelijk' met een BASIC-programma afgehandeld worden. Dat bleek de oplossing voor de gewenste combinatie van geluid en vloeiend bewegende sprites.

Naast de al genoemde 'stuiter-aanval' heeft Astro Challenge nog een aantal specialiteiten. De belangrijkste is een heuse 'challenging stage' waarin u, als beloning voor het afslaan van een aanval, de kans krijgt een aantal spookschepen te vernietigen zonder dat ze u kunnen raken.

Daarnaast bouwt het spel de achtergrond elke keer opnieuw 'random' op en bevat het een echte scoretabel die op schijf wordt bewaard.

Astro Challenge is dus niet alleen een spannend aktiespel, het biedt ook een schat aan procedures om te bestuderen.

★ Iban van der Wal



## Atari-ST TV-interface

**Let op: Dit mag u niet missen!!!**

**fl. 149,--**

(incl. BTW)

Ontdek nu de kleurrijke mogelijkheden die uw

Atari-ST biedt voor nog **geen 150 gulden**.

Met deze interface kunt u de Atari aansluiten op elke normale (kleuren) televisie en zo in kleur genieten van uw tekenprogramma's en spelletjes.

Binnen 5 minuten te installeren door:

- ingebouwde netvoeding
- aansluiting voor TV antenne ingang (kan. 36)
- composite video uitgang (cinch)
- audio uitgang (cinch)
- met Atari-monitor konnektor (Dus eenvoudig "inpluggen")

Levering uitsluitend onder rembours. Levering geschiedt volgens onze algemene leveringsvoorwaarden, gedeponeerd bij de KvK te Hengelo.

**-IBRUNELCO**

Bestelbon: uitknippen en in voldoende geïllustreerde envelop opsturen naar:  
Brunelco, Beethovenstraat 27  
7482 TE Haaksbergen

Bij deze bestel ik ..... Atari-TV interface(s)  
a f. 149,-- (incl. BTW, exd. f. 8,50 rembourskosten)

Ik verzoek u deze op te sturen naar:

Naam: .....

Adres: .....

Postcode: .....

Plaats: .....



```

Hidden
Spoke 1156,14
Spoke 16745024,0
For TX=1 To 28
  Read D.ata%
  Schot$=Schot$+Chr$(D.ata%)
Next TX
Data 0,0,1,0,2,0,3,0,4,0,5,0,6,25,7,0 =>
,8,16,9,16,10,16,11,0,12,10,13,9
For TX=1 To 28
  Read D.ata%
  Let Ontploft$=Ontploft$+Chr$(D.ata%)
Next TX
Data 0,0,1,0,2,0,3,0,4,0,5,0,6,10,7,0 =>
,8,16,9,16,10,16,11,0,12,80,13,9
Dim Naam$(6),PuntenZ(6)
If Exist("score.dat")
  Open "I",#1,"score.dat"
  For TX=1 To 5
    Input #1,Naam$(TX)
    Input #1,PuntenZ(TX)
  Next TX
  Close #1
Endif
Planeet$=Mki$(8)+Mki$(8)+Mki$(0)
Planeet$=Planeet$+Mki$(0)+Mki$(1)
For IX=1 To 16
  Read Voor,Achter
  Planeet$=Planeet$+Mki$(Achter)+Mki$ =>
(Voor)
Next IX
Data 0,0,0,2016,2016,6168,8184,9156,1 =>
1772,22498,16244,18410,28666,38885,23 =>
254,45033
Data 28666,38885,22390,43977,11244,21 =>
522,13652,19114,6840,9540,2016,6168,0 =>
,2016,0,0
Plof3$=Mki$(8)+Mki$(7)+Mki$(0)
Plof3$=Plof3$+Mki$(0)+Mki$(1)
For IX=1 To 16
  Read Voor,Achter
  Plof3$=Plof3$+Mki$(Achter)+Mki$(Voor)
Next IX
Data 2176,65535,8,65535,8320,65535,54 =>
5,65535,2056,65535,528,65535,8322,655 =>
35,33312,65535
Data 8322,65535,16,65535,2312,65535,3 =>
2769,65535,8736,65535,134,65535,8192, =>
65535,1288,65535
Plof2$=Mki$(8)+Mki$(7)+Mki$(0)
Plof2$=Plof2$+Mki$(0)+Mki$(1)
For IX=1 To 16
  Read Voor,Achter
  Plof2$=Plof2$+Mki$(Achter)+Mki$(Voor)
Next IX
Data 0,0,0,0,128,448,544,2032,2184,40 =>
88,864,8188,9362,16382,2600,16382
Data 9554,16382,544,8188,2504,4088,0, =>
2032,544,992,128,448,0,0,0,0
Plof1$=Mki$(8)+Mki$(7)+Mki$(0)
Plof1$=Plof1$+Mki$(0)+Mki$(1)
For IX=1 To 16
  Read Voor,Achter
  Plof1$=Plof1$+Mki$(Achter)+Mki$(Voor)
Next IX
Data 0,0,0,0,0,0,0,0,0,0,0,0,128,128,99 =>
2,544,2032
Data 320,992,0,448,0,0,0,0,0,0,0,0, =>
0,0,0
Let Ontploft$=Mki$(8)+Mki$(8)+Mki$(0)
Let Ontploft$=Ontploft$+Mki$(0)+Mki$(1)
For IX=1 To 16
  Read Voor,Achter
  Let Ontploft$=Ontploft$+Mki$(Achter =>
)+Mki$(Voor)
Next IX
Data 1088,0,0,0,2184,0,16384,0,34,0,5 =>
120,0,32842,0,0,0
Data 2321,0,32768,0,8194,0,544,0,1638 =>
8,0,4368,0,0,0,0,576,0
Vijand$=Mki$(8)+Mki$(7)+Mki$(0)
Vijand$=Vijand$+Mki$(0)+Mki$(1)
For IX=1 To 16
  Read Voor,Achter
  Vijand$=Vijand$+Mki$(Achter)+Mki$(V =>
oor)
Next IX
Data 0,0,0,0,0,0,0,0,0,2016,2016,6168 =>
,6168,10212,8196,24570
Data 32766,32769,65535,0,32766,32769, =>
8184,24582,0,8184,0,0,0,0,0,0
Speler$=Mki$(8)+Mki$(7)+Mki$(0)
Speler$=Speler$+Mki$(0)+Mki$(1)
For IX=1 To 16
  Read Voor,Achter
  Speler$=Speler$+Mki$(Achter)+Mki$(V =>
oor)
Next IX
Data 384,0,384,0,384,0,384,0,384,0,57 =>
6,0,576,0,576,0

```

```

Data 576,0,1440,0,3024,0,13932,0,2008 =>
2,0,49149,0,32769,0,65535,0
Speler1$=Speler$
Speler2$=Speler$
Planeet1$=Planeet$
Sprite Planeet$,Random(608)+16,Random =>
(305)+16
Sprite Planeet1$,Random(608)+16,Rando =>
m(305)+16
Sprite Speler1$,10,391
Sprite Speler2$,30,391
VixZ=Random(608)+16
ViyZ=Random(200)+16
Vijand1$=1
LevensZ=3
Op=6
StageZ=1
For T=1 To 200
  If Random(10)+1>1
    Plot Random(639)+1,Random(370)+1
  Else
    XZ=Random(639)
    YZ=Random(370)+1
    Draw XZ,YZ To XZ+1,YZ To XZ+1,YZ+ =>
1 To XZ,YZ+1 To XZ,YZ
  Endif
Next T
Line 0,370,639,370
Get 0,0,639,370,S$
Deftext 1,0,0,13
Text 540,398,"Score :"+Str$(ScoreZ)
Deffill 1,2,8
Box 150,100,490,300
Pbox 152,102,488,298
Deftext 0,16,0,32
Graphmode 2
Text 250,137,"Top Five"
Deftext 0,1,0,13
For YZ=1 To 5
  Text 170,150+25*YZ,YZ
  If Naam$(YZ)<>" "
    Text 190,150+25*YZ,Naam$(YZ)
  Else
    Text 190,150+25*YZ,String$(25,".")
  Endif
  If PuntenZ(YZ)>0
    Text 430,150+25*YZ,PuntenZ(YZ)
  Else
    Text 430,150+25*YZ,String$(5,".")
  Endif
Next YZ
Deftext 0,0,0,4
Text 208,290,"Press mouse button or k =>
ey to continue"
Repeat
  Until Len(Inkey$) Or Mousek
  Put 0,0,S$
  Deftext 1,0,0,13
  Graphmode 1
  While Mousek
  Wend
  Alert 0," Space Shot! =>
  |By Iban van der Wall! ",1,"Begin!! =>
",BZ
S:
Do
  Sprite Speler$,Mousex,360
  Gosub Vijand
  Gosub Geraakt
  If Mousek=1 And KogelZ=0
    Void Xbios(32,L:Varptr(Schot$))
    KogelZ=1
    Spr_xZ=Mousex
    Spr_yZ=350
    Line Spr_xZ,Spr_yZ,Spr_xZ,Spr_yZ-22
  Endif
  If KogelZ=1
    Color 0
    Line Spr_xZ,Spr_yZ,Spr_xZ,Spr_yZ- =>
Int(Op)
    Color 1
    Line Spr_xZ,Spr_yZ-22,Spr_xZ,Spr_ =>
yz-22-Int(Op)
    Spr_yZ=Spr_yZ-Int(Op)
    If Spr_yZ<=-1
      KogelZ=0
      Spr_xZ=9999
      Spr_yZ=9999
    Endif
  Endif
Exit If HitZ>0
Loop
Score1Z=Int((400-ViyZ)/10)*(Int(Op)-5)
VixZ=Random(608)+16
ViyZ=Random(200)+16
RZ=Random(2)
BZ=Random(2)
Spr_xZ=9999
Spr_yZ=9999
KogelZ=0

```

```

If HitZ=1
  ScoreZ=ScoreZ+Score1Z
  If Op+0.3<=21
    Op=Op+0.3
  Endif
  Inc Kna1Z
  If Kna1Z=10
    Kna1Z=0
    If StageZ+1<=5
      Inc StageZ
      Verhoging!=True
    Else
      Verhoging!=False
    Endif
    Gosub Bonus
    While Mousek
    Wend
    If Verhoging!
      Alert 0," Stage "+Str$(StageZ =>
)+ " ",1," Ok ",BZ
    Endif
  Endif
Endif
HitZ=0
Put 0,0,S$
Text 540,398,"Score :"+Str$(ScoreZ)
Put 0,370,Ships$
Goto 5
Procedure Vijand
  If Vijand1Z=1
    Vsync
    Sprite Vijand$,VixZ,ViyZ
    If RZ=1
      VixZ=VixZ+Int(Op)
      If VixZ>=623
        VixZ=623
        RZ=0
      Endif
      If BZ=1
        ViyZ=ViyZ-Int(Op/2)
        If ViyZ<=16
          ViyZ=16
          BZ=0
        Endif
      Else
        ViyZ=ViyZ+Int(Op/2)
        If ViyZ>=360
          ViyZ=360
          BZ=1
        Endif
      Endif
    Else
      VixZ=VixZ-Int(Op)
      If VixZ<=16
        VixZ=16
        RZ=1
      Endif
      If BZ=1
        ViyZ=ViyZ-Int(Op/2)
        If ViyZ<=16
          ViyZ=16
          BZ=0
        Endif
      Else
        ViyZ=ViyZ+Int(Op/2)
        If ViyZ>=360
          ViyZ=360
          BZ=1
        Endif
      Endif
    Endif
  Endif
Endif
Return
Procedure Geraakt
  If Abs(Spr_xZ-VixZ)<=18 And Abs(Spr =>
_yZ-ViyZ)<=18
    Void Xbios(32,L:Varptr(Ontploft$))
    Color 0
    Line Spr_xZ,Spr_yZ,Spr_xZ,Spr_yZ-22
    Color 1
    Spoke 16745024,1
    Pause 3
    Spoke 16745024,0
    Sprite Vijand$
    Sprite Plof1$,VixZ,ViyZ
    Pause 8
    Sprite Plof1$
    Sprite Plof2$,VixZ,ViyZ
    Pause 8
    Sprite Plof2$
    Sprite Plof3$,VixZ,ViyZ
    Pause 8
    Put 0,0,S$
    HitZ=1
  Endif
  If Abs(Mousex-VixZ)<=18 And Abs(360 =>
-ViyZ)<=18

```



# SALES REPRESENTATIVE ATARI/AMIGA m/v



## De functie-inhoud

- ✓ Zelfstandig prospectus behandelen.
- ✓ Demonstraties verzorgen.
- ✓ Orders afsluiten.

## De functie-eisen

- ✓ Goede contactuele eigenschappen.
- ✓ Representatief voorkomen.
- ✓ Natuurlijk overwicht.
- ✓ Ervaring met computers (Atari en/of Commodore) strekt tot aanbeveling.

Wij bieden een interessante baan in een dynamische markt. Wij elken van onze medewerkers, dat ze zelfstandig kunnen werken en geven een grote mate van vrijheid. De primaire en secundaire arbeidsvoorwaarden liggen boven het gemiddelde. Kandidaten die samen met ons aan een goede toekomst willen werken nodigen wij uit schriftelijk te reageren t.a.v. de heer N. Steenhuis.

Compudress bv. Postbus 6-3470 CA Kamerik (bij Woerden) - Tel. 03461-2124

```

Void Xbios(32,L:Varptr(Ontplof$))
X_pos=Mousex
Spoke 16745024,1
Pause 3
Spoke 16745024,0
Put 0,0,S$
Sprite Speler$
Sprite Vijand$
Sprite Plof1$,X_pos,360
Pause 8
Sprite Plof1$
Sprite Plof2$,X_pos,360
Pause 8
Sprite Plof2$
Sprite Plof3$,X_pos,360
Pause 8
Sprite Plof3$
Put 0,0,S$
Hit%=2
Levens%=Levens%-1
If Levens%=2
Wave 0,0
Sprite Speler2$
Alert 3,"Second ship....",1,"Su =>
ccess",BZ
Get 0,370,100,399,Ships$
Endif
If Levens%=1
Wave 0,0
Sprite Speler1$
Alert 3,"Last Ship....",1,"Succ =>
ess",BZ
Get 0,370,100,399,Ships$
Endif
If Levens%=0
Wave 0,0
Put 0,0,S$
If Score%>0
Gosub Inschrijven
Endif
Alert 2,"Do you want to play?ag =>
ain?",1,"Yes!No",BZ
If BZ=1
Run
Else
Cls
Spoke 16745024,1
Spoke 1156,15
Wave 0,0
Edit
Endif
Endif
Return

Procedure Inschrijven
Ja%=0
For TZ=1 To 5
Exit If Score%>Punten%(TZ)
Next TZ
Ja%=TZ
If Ja%>0 And Ja%<6
For TZ=5 Downto Ja%
Naam$(TZ+1)=Naam$(TZ)
Punten%(TZ+1)=Punten%(TZ)
Next TZ
Naam$(Ja%)=""
Punten%(Ja%)=Score%
Deffill 1,2,8
Box 150,100,490,300
Pbox 152,102,488,298
Deftext 0,16,0,32
Graphmode 2
Text 250,137,"Top Five"
Deftext 0,1,0,13
For YZ=1 To 5
Text 170,150+25*YZ,YZ
If Naam$(YZ)<>""
Text 190,150+25*YZ,Naam$(YZ)
Else
If YZ<Ja%
Text 190,150+25*YZ,String$( =>
25,".")
Endif
Endif
If Punten%(YZ)>0
Text 430,150+25*YZ,Punten%(YZ)
Else
Text 430,150+25*YZ,String$(5, =>
".")
Endif
Next YZ
Deftext 0,0,0,4
Text 223,290,"Enter your name and =>
press <return>"
Deftext 0,1,0,13
Do
Toets%=Inp(2)
Exit If Toets%=13
If Toets%=8 And Len(Naam$(Ja%))>0

```

```

Naam$(Ja%)=Left$(Naam$(Ja%),L =>
en(Naam$(Ja%))-1)
Endif
If Toets%<>8 And Len(Naam$(Ja%) =>
)<25
Naam$(Ja%)=Naam$(Ja%)+Chr$(To =>
ets%)
Endif
Text 190,150+25*Ja%,Naam$(Ja%)
Pbox 190+Len(Naam$(Ja%))*8,152+ =>
25*Ja%,191+Len(Naam$(Ja%))*8+8,152+25 =>
*Ja%-16
Loop
Put 0,0,S$
Open "0",#1,"score.dat"
For TZ=1 To 5
Print #1,Naam$(TZ)
Print #1,Punten%(TZ)
Next TZ
Close #1
Endif
Return

Procedure Bonus
Wave 0,0
Put 0,0,S$
Vix%=0
Deftext 1,1,0,32
For TZ=1 To 8
Graphmode 4
Text 100,200,440,"CHALLENGING STAGE"
For T1Z=1 To 8
Sound 1,15,T1Z,4,1
Next T1Z
Graphmode 1
Text 100,200,440,"CHALLENGING STAGE"
For T1Z=1 To 8
Sound 1,15,T1Z,4,1
Next T1Z
Next TZ
Wave 0,0
Pause 50
Deftext 1,0,0,13
Put 0,0,S$
Do
Sprite Speler$,Mousex,360
Vsync
Sprite Vijand$,Vix%,175
If Mousex=1 And Kogel%=0
Void Xbios(32,L:Varptr(Schot$))
Kogel%=1
Spr_x%=Mousex
Spr_y%=350
Line Spr_x%,Spr_y%,Spr_x%,Spr_y =>
X-22
Endif
If Kogel%=1
Color 0
Line Spr_x%,Spr_y%,Spr_x%,Spr_y =>
%-Int(Op)
Color 1
Line Spr_x%,Spr_y%-22,Spr_x%,Sp =>
r_y%-22-Int(Op)
Spr_y%=Spr_y%-Int(Op)

```

```

If Spr_y%<=-1
Kogel%=0
Spr_x%=9999
Spr_y%=9999
Endif
Endif
If Abs(Spr_x%-Vix%)<=18 And Spr_y =>
Z-11>166 And Spr_y%-11<184
Void Xbios(32,L:Varptr(Ontplof$))
Color 0
Line Spr_x%,Spr_y%,Spr_x%,Spr_y =>
Z-22
Color 1
Inc Aantal_hit%
Kogel%=0
Spr_x%=9999
Spr_y%=9999
Spoke 16745024,1
Pause 3
Spoke 16745024,0
Put 0,0,S$
Sprite Vijand$
Sprite Plof1$,Vix%,175
Pause 8
Sprite Plof1$
Sprite Plof2$,Vix%,175
Pause 8
Sprite Plof2$
Sprite Plof3$,Vix%,175
Pause 8
Sprite Plof3$
Put 0,0,S$
Vix%=639
Endif
Vix%=Vix%+Int(Op)-4
If Vix%>639
Put 0,0,S$
Vix%=0
Inc Keer%
Endif
Exit If Keer%>20 Or Aantal_hit%=10
Loop
Put 0,0,S$
If (Aantal_hit%/Keer%)*100=>75
Extra%=5000
Else
If (Aantal_hit%/Keer%)>25
Extra%=1000
Else
Extra%=0
Endif
Endif
Wave 0,0
Alert 0,"You scored "+Str$(Aantal_h =>
it%)+ " hits out of a possible "+Str$( =>
Keer%)+ "!!! Bonuspoints : "+Str$(Extra% =>
)+ "!",1," Ok ",BZ
Score%=Score%+Extra%
Keer%=0
Aantal_hit%=0
Kogel%=0
Spr_x%=9999
Spr_y%=9999
Return

```



# MACHINETAAL OP DE ST

## DEEL 5: LINE A ROUTINES

Elders in dit nummer vindt u een artikel van Frank Abbing over "de munitie" van de ST. Dit vormt een prima aanvulling op het onderwerp dat de vorige keer in deze rubriek behandeld werd: de excepties. Daarmee is dat overzicht compleet en kunnen we ons bezig gaan houden met het restant van de machinetaal interrupts.

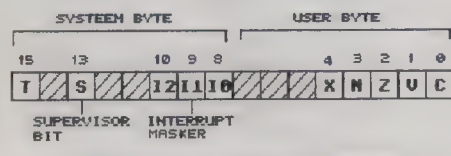
Tot slot van deze aflevering vindt u een voorbeeld waarin de zogenaamde 'line A routines' gebruikt worden, die allerlei grafische effecten mogelijk maken.

### INTERRUPT BEHANDELING

Wat een interrupt is en doet werd uitgebreid behandeld in START 5. Hoe de 68000 interrupts verwerkt, wordt hierna uit de doeken gedaan.

In figuur 1 zien we hoe het statusregister opgebouwd is. De drie bits I2, I1 en I0 vormen samen het interruptmasker. Deze bits zijn in staat de acht interrupt niveaus uit tabel 1 te creëren. Opgemerkt moet worden dat de externe lijnen IPL2, IPL1 en IPL0 in vergelijking met I2, I1 en I0 een geïnverteerd niveau toegevoerd krijgen.

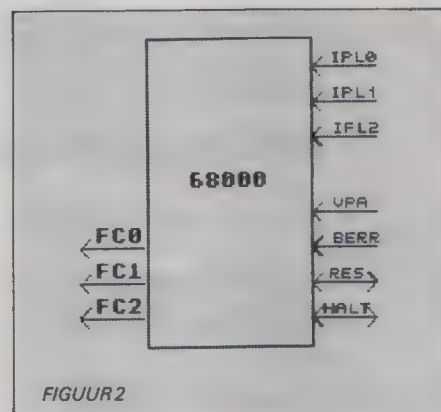
FIGUUR 1



De 68000 inverteert deze externe lijnen voor een juiste vergelijking met het interruptmasker (zie figuur 2). Als er een interrupt aanvraag plaatsvindt, vergelijkt de 68000 zijn ingangslijnen met de bits van het interruptmasker in het status register. Na het beëindigen van een lopende instructie bekijkt de processor de interrupt. Is deze lager of gelijk aan het niveau in het interrupt masker, dan wordt de interrupt niet gehonoreerd. Is hij daarentegen hoger, dan wordt het exceptie-proces voor interrupts gestart.

Een belangrijk onderdeel in de exceptie-cyclus is het hoog (op '1') zetten van de bits in het interruptmasker die zich op het zelfde niveau als de aangevraagde interrupt bevinden. Hiermee wordt voorkomen dat lagere interrupts de verwerking van de exceptie onderbreken. Hogere prioriteiten worden uiteraard wel gehonoreerd. Daarna haalt de processor het vektornummer, dat door de aanvrager van de interrupt gegenereerd is, van de databus en verwerkt dit tot een vektoradres. Door middel van dit vektoradres vinden we uiteindelijk in de exceptietabel het adres waarmee de programmateller geladen moet worden om de gewenste interruptroutine te bereiken.

Een andere mogelijkheid wordt gevormd door de autovektoren. De exceptietabel uit START 5 geeft zeven lokaties met autovektoren aan, die overeenkomen met de zeven interruptniveaus. De vektoren worden alleen gebruikt als de zogenaamde VPA lijn (zie figuur 2) door de aanvrager van de interrupt actief is gemaakt. De processor haalt in dat geval niet het vektornummer van de bus, maar gebruikt de autovektor uit de tabel die overeenkomt met het interruptniveau.



Nu zijn alleen de interruptniveaus zeven en nul uit tabel 1 nog niet besproken. Het laatste geeft aan: geen interrupt aanvragen. Het allerhoogste (zeven) niveau geeft een niet maskeerbaar interrupt. Dit komt overeen met de NMI zoals we die van de 6502 kennen.

### VOORBEELD PROGRAMMA

Ons voorbeeld programma maakt gebruik van 'line A routines'. De 68000 kent een tweetal excepties die betrekking hebben op "niet bestaande" opcodes. Deze zijn \$AXXX en \$Fxxx, waarbij alleen de hoogste vier bits de vorm van de exceptie bepalen. De ATARI ST gebruikt de opcode \$AXXX om grafische routines te implementeren. De laatste nibble in \$AXXX selecteert een van de 14 grafische routines. In het programma worden de volgende gebruikt:

**\$A000:** Met deze aanroep verkrijgen we het adres van een data gebied waarin onder andere de coördinaten van de punten waartussen een lijn getrokken moet worden thuis horen.

**\$A003:** Deze routine trekt een lijn tussen de coördinaten x1,y1 en x2,y2. De coördinaten moeten, zoals bij \$A000 reeds vermeld is, van te voren opgegeven zijn.

Uiteindelijk zal het programma het woord op het scherm toveren, waarbij de letters met behulp van losse lijnstukken

getekend zijn. De belangrijkste onderdelen van het programma zijn:

**START:** Het programma stuurt de cursor naar de home-positie en wist het scherm. De subroutine SCHCH verricht het werk.

**VARZET:** 'init' zet het startadres van het gebied waar het programma de coördinaten van de letters moet vinden, in adresregister A0. De variabelen die voor de goede werking van het programma van belang zijn, worden daarna in dit deel van het geheugen opgeslagen.

**HLIJN:** Allereerst worden de drie horizontale lijnstukken van de eerste E getrokken. In figuur 3 zien we dat de eerste lijn van de "E" met behulp van de coördinaten x1,x2 en y1 tot stand komt. Door y1 in een lus met een konstante waarde te vermeerderen, zijn de twee andere horizontale lijnstukken van de "E" ook te trekken. Om dit gemakkelijk te maken, worden de coördinaten eerst in registers geladen.

Voor de tweede "E" is eveneens een lus opgezet. Deze corrigeert alleen de x1 en x2 coördinaten en duikt vervolgens de eerste lus in.

TABEL 1

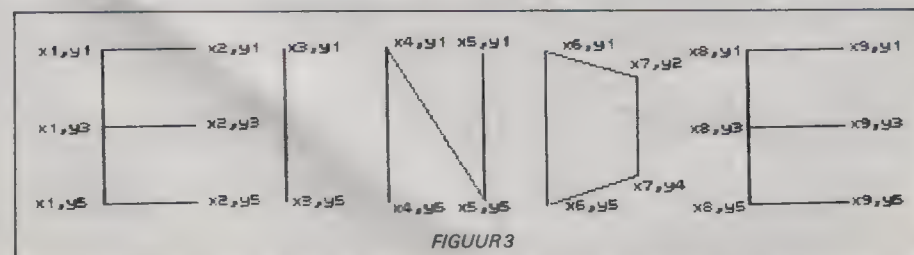
NIVEAU	I2	I1	I0	WERKING
7	1	1	1	MASKEERBAAR INTERRUPT
6	1	1	0	
5	1	0	1	
4	1	0	0	
3	0	1	1	
2	0	1	0	
1	0	0	1	
0	0	0	0	GEEN INTERRUPT

**VLIJN:** Met de verticale lijnen halen we net zo'n spelletje uit. Zo lang de afstanden maar konstant blijven, kunnen we snel werken. Eerst worden de verticale lijnen van de "E" en "I" getrokken. Dan volgt een correctie om juist uit te komen voor de eerste lijn van de "N" en tenslotte worden de verticale lijnen van de "D" en de "E" getrokken.

**ONTBRE:** De verticale lijnen die niet goed in het VLIJN algoritme pasten en de ontbrekende diagonale lijnen van de "D" worden verzorgd door een macro aanroep. In de aanroep worden de coördinaten (vier stuks) opgegeven en de macro DIAG (bovenaan in het programma) trekt de ontbrekende lijnen.

**TOETSIN:** Met een CR (return) keren we terug naar de desktop.

Jan Vogel



FIGUUR 3



```

fg_bp_1 equ 24
fg_bp_2 equ 26
fg_bp_3 equ 28
fg_bp_4 equ 30
lstlin equ 32
ln_mask equ 34
wrt_mode equ 36
x1 equ 38
y1 equ 40
x2 equ 42
y2 equ 44
init equ $a000
lijn equ $a003
yafst equ 50
e2afst equ 190
xafst equ 60

macro diag
    move \1,x1(a0)
    move \2,y1(a0)
    move \3,x2(a0)
    move \4,y2(a0)
    move a0,a4
    dc.w lijn
    move a4,a0
endm

start:
move #27,-(a7)
schsch
move #72,-(a7)
schsch
move #27,-(a7)
schsch
move #74,-(a7)
schsch
move #0,CCR
dc.w init
move #1,fg_bp_1(a0)
clr fg_bp_2(a0)
clr fg_bp_3(a0)
clr fg_bp_4(a0)
move #-1,lstlin(a0)
move #-1,ln_mask(a0)
move #0,wrt_mode(a0)
movea.l #horz,a5
move (a5)+,d7
move (a5)+,d6
move (a5)+,d5
move #2,d4
moveq #1,d3
move d7,x1(a0)
move d6,y1(a0)
move d7,x2(a0)
move d5,y2(a0)
move a0,a4
dc.w lijn
move a4,a0
add #xafst,d7
dbra d4,vlijn
sub #50,d7
moveq #2,d4
dbra d3,vlijn
ontbre:
diag #320,#150,#320,#250
diag #270,#150,#320,#250
diag #330,#150,#380,#175
diag #330,#250,#380,#225
diag #380,#175,#380,#225
jmp toetsin
schsch:
link a6,#0
movea.l 8(a6),-(a7)
move #2,-(a7)
trap #1
unlk a6
rts

toetsin:
move #1,-(a7)
trap #1
addq.l #2,a7
cmpl.b #50,d0
beq desktop
jmp toetsin
move #0,-(a7)
trap #1
horz:
dc.w 200,150,250
vert:
dc.w 200,150,250
end

;
;Met dit voorbeeld trekken we echt
;de lijn en wel door het woordje
;"EINDE" op het scherm te schrijven
;met behulp van de "line A"
;routines.
;Met de toets CR keren we terug
;naar de desktop.

;adres variabelen line A reddden
;de lijn trekken
;adres variabelen line A terugzetten
;Y correctie voor drie streepjes
;weer een E streep
;X afstand tweede E corrigeren
;Y coördinaat op aanvangs waarde
;drie streepjes
;twee E's
;adreswijzer van vertikale lijnen
;registers laden met coördinaten
;eerst twee vertikale lijnen
;coördinaten laden
;adres variabelen reddden
;de lijn trekken
;adres variabelen herstellen
;juiste afstand tussen vert. lijnen
;eerst twee vertikale lijnen
;X correctie tweede groep v lijnen
;na tweemaal vert. lijnen stoppen
;ontbrekende lijn N-letter
;diagonaal N-letter
;bovenste diag. D-letter
;onderste diag. D-letter
;achterste lijn D-letter
;wacht op CR
;save oude stackpointer
;zet karakter op juiste plaats
;gemdos kode uit
;herstel oude stackpointer
;gemdos kode in
;stack correctie
;CR?
;programma verlaten
;alleen op CR reageren
;gem desk
;horizontale strepen E
;de meeste vertikale lijnen

```





# DE MUNITIE VAN DE ST

Het TOS van de Atari-ST toont soms bommen op het scherm om aan te geven dat er in het systeem iets fout is gegaan. Het is niet erg duidelijk wat er fout is, of wat het aantal bommen betekent. Soms krijg je bovendien de tijd niet om te tellen hoeveel bommen er vallen. Frank Abbing verklaart in dit artikel de betekenis van de bommeldingen. Bovendien krijgt u een handig programma kado dat het visueel geweld omzet in overzichtelijke tekstinformatie.

## BOMMEN EN HUN DOEL

Onderzoek van het TOS en bestudering van de 68000 CPU maken duidelijk dat

bommen een uiting zijn van 'exceptions'. Een 'exception' in de 68000 is alles wat niet tot de normale afloop van een programma behoort. Sommige van die exceptions worden door het TOS wel als 'normaal' beschouwd. TOS heeft er een afhandelingsroutine voor. Sommige andere worden als fouten beschouwd: dan gaat TOS met bommen gooien. De belangrijkste exceptions zijn:

0 (en 1) Komen in actie bij reset. Dit wordt door TOS verwerkt.

2 Bus Error. Het gebruik van adressen boven 4 megabyte begrijpt de Atari hardware niet. Adressen onder de \$800 zijn in 'user-mode' beschermd. Die geven ook een "Bus Error" als geprobeerd wordt daar iets te doen.

3 Address Error. Als een oneven adres niet voor een byte maar voor een woord of een lang woord gebruikt wordt, is er sprake van een "address error".

4 Illegal Instruction. Niet alle bitcombinaties zijn 68000 instructiekodes. Wordt zo'n code toch gebruikt, dan geeft dat een 'Illegal Instruction Exception'. Meestal wordt dit veroorzaakt door een niet goed afgesloten programma: het programma sukkelde van het programma- in het datageheugen.

5 Zero Divide. De 68000 vindt delen door nul heel erg. TOS vindt dat niet. Als 'afhandeling' dient de RTE (ReTurn from Exception). Het programma moet maar verder met een niet gedeeld deeltal, vindt TOS. Geen 5 bommen dus.

Om de een of andere reden zit er een Zero Divide in het programma 'Textomat'. Die treedt op bij het schoonmaken van het werkgeheugen. Onder TOS heb je daar geen last van. Een programma dat die Zero Divide opvangt, verstoort 'Textomat'! (Textomat mag dus niet met het programma uit dit artikel gebruikt worden!)

6 CHK instruction. De CHK instructie is bedoeld om vast te stellen of de waarde van een indexregister zich binnen bepaalde grenzen bevindt. Is die waarde buiten de grenzen, dan treedt exception 6 op.

7 TRAPV instruction. Voert een programma een TRAPV uit, dan zal deze exception optreden als het V-bit aanstaat. Het V-bit is het overflow bit. Met andere woorden, dit is een 'trap-if-overflow'.

8 Privilege violation. Sommige instructies kunnen alleen in Supervisor mode worden uitgevoerd. In de (normale) User mode geven deze exception nummer 8.

9 Trace. Die zal in een 'gewoon' programma niet voorkomen. Iemand die voldoende kennis van zaken heeft om een trace exception te veroorzaken zal ook wel weten wat ermee te doen. (Bijvoorbeeld 'single step').

10 Line A. Die geeft geen bommen, omdat TOS deze exception voor de 'line A' routines gebruikt. Dat zijn heel snelle grafische TOS-routines waarmee heel gemakkelijk lijnen, rechthoeken en cirkels getekend kunnen worden. Vanuit een assembler programma zijn die heel goed aan te roepen.

Het lijstje is nog niet klaar. We hebben de exceptions 11 tot en met 255 nog over. De meeste daarvan hebben te maken met 'interrupt'. Dat is een signaal dat meestal niet door een verkeerd programma wordt opgewekt. Een melding van 200 bommen zal daarom niet voorkomen.

## BOMMENTOLK

De meest voorkomende bommeldingen zijn 2, 3, 4 of 8 bommen. Die worden opgevangen door het bijgevoegde programma. Het programma geeft een verklarende tekst en een 'dump' van alle ter beschikking staande inhoud van de adres- en dataregisters van de 68000. Daarna wacht het op een toetsdruk alvorens naar de desktop terug te keren. Soms is er iets grondig fout en lukt de terugkeer niet. Reset is dan de enige oplossing, maar dat is bij de TOS-bommeldingen niet anders.

Het programma kan in de AUTO-folder gezet worden. Het is kleiner dan 1K.

Als u het programma start gebeurt er ogenschijnlijk niets. Het heeft fouten nodig om actief te worden. De bijgevoegde vier kleine assemblerprogramma's veroorzaken exceptions waarmee u de goede werking van uw utility kunt testen.

Het commentaar in de listing bevat een aantal malen getallen tussen haakjes. Deze cijfers verwijzen naar bladzijden in het boek "Atari ST Intern" van Data Becker Nederlands.

```
*****
* bomldng.s
* dit programma komt in actie bij 'exception'
* het geeft een leesbare boodschap
* return brengt desktop terug
* f. r. abbing
*****
```

```
start    moveq    #2,d3          vector nummer
          lea      vectab(pc),a3
setvec    move.l    (a3)+,-(sp)
          move.w    d3,-(sp)      (147)
          move.w    #5,-(sp)
          trap      #13
          addq.l    #8,sp
no_set    addq.w    #1,d3
          cmp.w     #10,d3        line 1010
          blt       setvec
          clr.w     -(sp)
          move.l    #1500,-(sp)    lengte bomldng.prg
          move.w    #31,-(sp)      keep process (119)
          trap      #1
```

```
v2        bsr      savregs
          lea      busmes(pc),a0
          bra.s     v2_3
```

```
v3        bsr      savregs
          lea      adrmes(pc),a0
v2_3      bsr      linout
          lea      locmes(pc),a0
          bsr      linout
          move.l    2(sp),d0      foute locatie
          bsr      hexout8
          move.w    (sp),d0      foutcode
*bit 4 write/read, 3 instruction/not
          btst     #4,d0
          bne      bit4on
          lea      wrtmes(pc),a0
bit4off    bsr      linout
          and.w     #7,d0          1,2,5,6 of 7
          cmp.w     #7,d0
          beq      intack
          btst     #2,d0
          bne      super
          lea      usrmes(pc),a0
user       bsr      linout
          btst     #0,d0
          beq      progr
          lea      datmes(pc),a0
noprg      bsr      linout
          lea      insmes(pc),a0
          bsr      linout
```







```
* buserror.s
```

```
start moveq #-100,d0      5 goeie en een foute
      moveq #-101,d1
      moveq #-102,d2
      move.l #'ABCD',d3
      move.l #$ABCD,d4
*      move.b 100,d5      <--verboden gebied!
      move.b 5000000,d5    Buiten bereik ST
      end
```

```
* illegal.s
```

```
start move.l #$abcdef,a0
      move.w #$8123,a1
      move.w a0,a2
      move.w a1,a3
*      die hierboven zijn goed, al doen ze misschien
*      niet wat je denkt. Dit is 'ILLEGAL':
      dc.w $4afc
      end
```

```
* zerdiv.s
```

```
start divs #0,d1
      end
```



```
Data 96,26,0,0,3,148,0,0,0,0
Data 0,0,0,0,0,0,0,0,0,0
Data 0,0,0,0,0,0,0,118,2
Data 71,250,1,196,47,27,63,3,63,60
Data 0,5,78,77,80,143,82,67,12,67
Data 0,10,109,236,66,103,47,60,0,0
Data 5,220,63,60,0,49,78,65,97,0
Data 1,14,65,250,2,137,96,8,97,0
Data 1,4,65,250,2,135,97,0,1,60
Data 65,250,2,139,97,0,1,52,32,47
Data 0,2,97,0,1,70,48,23,8,0
Data 0,4,102,0,0,194,65,250,2,194
Data 97,0,1,26,2,64,0,7,12,64
Data 0,7,103,0,0,182,8,0,0,2
Data 102,0,0,182,65,250,2,187,97,0
Data 0,254,8,0,0,103,0,0,174
Data 65,250,2,189,97,0,0,238,65,250
Data 2,79,97,0,0,230,48,47,0,6
Data 97,0,0,238,80,143,65,250,2,78
Data 97,0,0,212,48,23,97,0,0,222
Data 65,250,2,85,97,0,0,198,32,47
Data 0,2,97,0,0,216,65,250,2,60
Data 97,0,0,182,78,107,32,11,97,0
Data 0,200,8,128,0,0,38,64,78,99
Data 65,250,1,128,97,0,0,158,67,250
Data 2,116,116,1,118,7,32,25,97,0
Data 0,170,81,203,255,248,74,130,103,8
Data 65,250,1,186,97,0,0,128,81,202
Data 255,230,63,60,0,1,78,65,84,143
Data 62,188,3,0,47,124,0,0,1,20
Data 0,2,78,115,66,103,78,65,65,250
Data 2,13,96,0,255,62,65,250,1,235
Data 96,0,255,98,65,250,2,13,96,0
Data 255,74,65,250,2,23,96,0,255,82
Data 72,249,255,255,0,0,3,82,78,117
Data 97,244,65,250,0,178,96,38,97,236
Data 65,250,0,182,96,30,97,228,65,250
Data 0,190,96,22,97,220,65,250,0,197
Data 96,14,97,212,65,250,0,206,96,6
Data 97,204,65,250,0,222,97,0,0,6
Data 96,0,255,40,72,231,240,0,72,80
Data 63,60,0,9,78,65,92,143,96,42
Data 72,231,240,0,118,3,72,64,96,6
Data 72,231,240,0,118,7,34,0,233,153
Data 20,1,2,66,0,15,16,59,32,68
Data 97,16,81,203,255,240,16,60,0,32
Data 97,4,76,223,0,15,78,117,2,64
Data 0,127,63,0,63,60,0,2,78,65
Data 88,143,78,117,0,0,0,40,0,0
Data 0,50,0,0,1,66,0,0,1,74
Data 0,0,1,82,0,0,1,90,0,0
Data 1,98,0,0,1,106,48,49,50,51
Data 52,53,54,55,56,57,97,98,99,100
Data 101,102,10,13,27,75,73,108,108,101
Data 103,97,108,0,10,13,27,75,90,101
Data 114,111,32,68,105,118,105,100,101,0
Data 10,13,27,75,67,72,75,32,105,110
Data 115,116,114,46,0,10,13,27,75,84
Data 82,65,80,86,32,105,110,115,116,114
Data 46,0,10,13,27,75,80,114,105,118
Data 105,108,101,103,101,32,86,105,111,108
Data 97,116,105,111,110,0,10,13,27,75
Data 84,114,97,99,101,0,10,13,27,75
Data 32,32,32,32,32,32,32,32,32,32
Data 48,32,32,32,32,32,32,32,32,49
Data 32,32,32,32,32,32,32,32,50,32
Data 32,32,32,32,32,32,32,51,32,32
Data 32,32,32,32,32,32,52,32,32,32
```

```
Data 32,32,32,32,32,53,32,32,32,32
Data 32,32,32,32,54,32,32,32,32,32
Data 32,32,32,55,32,10,13,27,75,68
Data 110,58,32,0,10,13,27,75,65,110
Data 58,32,0,10,13,27,75,66,85,83
Data 0,10,13,27,75,65,68,68,82,69
Data 83,83,0,32,101,114,114,111,114,32
Data 111,112,32,97,100,114,101,115,58,32
Data 0,10,13,27,75,73,110,115,116,114
Data 117,99,116,105,101,58,32,0,32,32
Data 32,83,116,97,116,117,115,58,32,0
Data 44,32,32,85,83,80,58,32,0,32
Data 32,32,80,46,67,46,58,32,0,10
Data 13,27,75,73,110,116,46,32,65,99
Data 107,110,46,0,10,13,27,75,119,114
Data 105,116,101,32,0,10,13,27,75,114
Data 101,97,100,32,0,117,115,101,114,32
Data 0,115,117,112,101,114,118,105,115,111
Data 114,32,0,100,97,116,97,46,0,112
Data 114,111,103,114,97,109,46,0,0,0
Data 0,0,0,0,0,0,0,0,0,0
Data 0,0,0,0,0,0,0,0,0,0
Data 0,0,0,0,0,0,0,0,0,0
Data 0,0,0,0,0,0,0,0,0,0
Data 0,0,0,0,0,0,0,0,0,0
Data 0,0,0,0,0,0,0,0,0,0
Data 0,0,0,0,0,0,1,12,48,140
Data 4,4,4,4,4,4,4,4,4,4
Open "0",#1,"bom.prg"
For I=1 To 958
  Read A
  B=B+A
  Print #1,Chr$(A);
Next I
Close #1
If B<>59257
  Print "Er zit een fout in een dataregel!"
Endif
```

```
Data 96,26,0,0,0,24,0,0,0,0,0
Data 0,0,0,0,0,0,0,0,0,0
Data 0,0,0,0,0,0,0,0,112,100
Data 114,101,116,102,38,60,65,66,67,68
Data 40,60,0,0,171,205,42,57,0,0
Data 171,205,0,0,0,0,0,0,0,0
Open "0",#1,"adres.prg"
For I=1 To 60
  Read A
  B=B+A
  Print #1,Chr$(A);
Next I
Close #1
If B<>2106
  Print "Er zit een fout in een dataregel!"
Endif
```

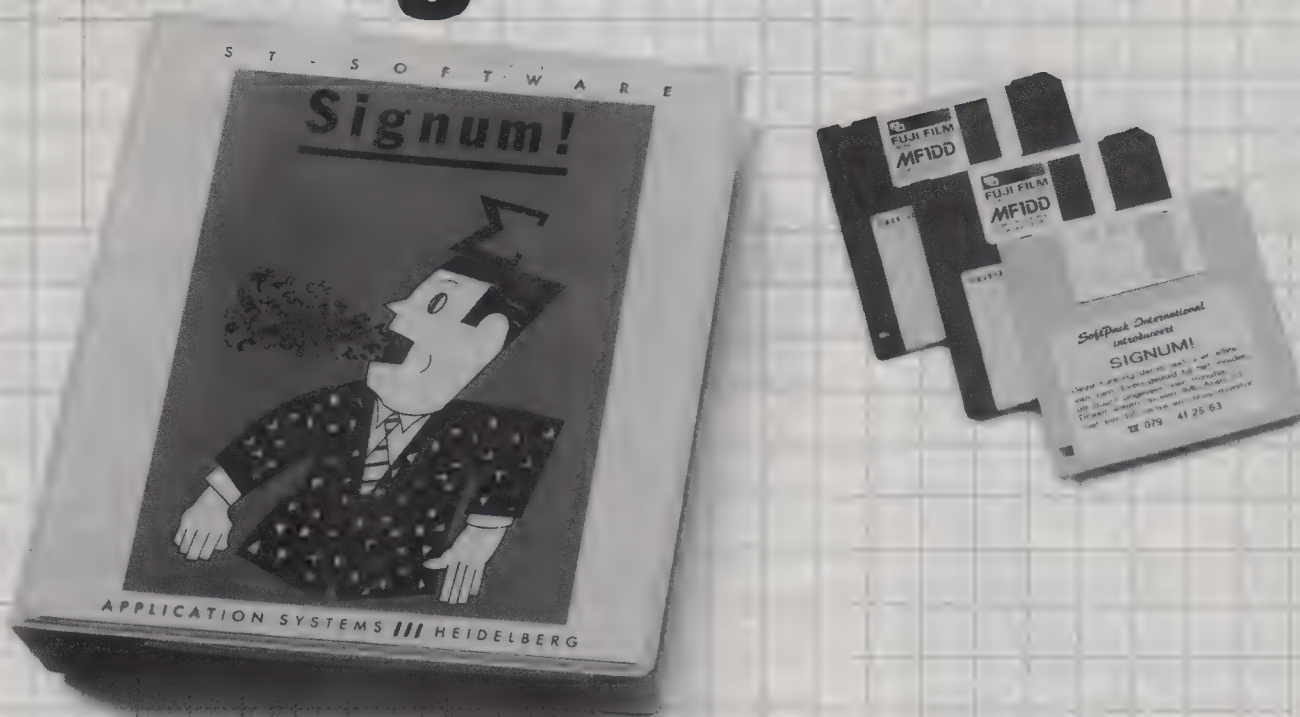
```
Data 96,26,0,0,0,24,0,0,0,0,0
Data 0,0,0,0,0,0,0,0,0,0
Data 0,0,0,0,0,0,0,0,112,156
Data 114,155,116,154,38,60,65,66,67,68
Data 40,60,0,0,171,205,26,57,0,76
Data 75,64,0,0,0,0,0,0,0,0
Open "0",#1,"bus.prg"
For I=1 To 60
  Read A
  B=B+A
  Print #1,Chr$(A);
Next I
Close #1
If B<>2091
  Print "Er zit een fout in een dataregel!"
Endif
```

```
Data 96,26,0,0,0,16,0,0,0,0,0
Data 0,0,0,0,0,0,0,0,0,0
Data 0,0,0,0,0,0,0,0,32,124
Data 0,171,205,239,50,124,129,35,52,72
Data 54,73,74,252,0,0,0,0,0,0
Data 0,0
Open "0",#1,"illegaal.prg"
For I=1 To 52
  Read A
  B=B+A
  Print #1,Chr$(A);
Next I
Close #1
If B<>1824
  Print "Er zit een fout in een dataregel!"
Endif
```

```
Data 96,26,0,0,0,4,0,0,0,0,0
Data 0,0,0,0,0,0,0,0,0,0
Data 0,0,0,0,0,0,0,0,131,252
Data 0,0,0,0,0,0,0,0,0,0
Open "0",#1,"nuldeel.prg"
For I=1 To 40
  Read A
  B=B+A
  Print #1,Chr$(A);
Next I
Close #1
If B<>509
  Print "Er zit een fout in een dataregel!"
Endif
```



# Het gouden trio!



Dat is de Atari ST 1040, de Nec-Pinwriter serie en het tekstverwerkingspakket SIGNUM!

Bij een perfecte computer en dito printer hoort een softwarepakket dat alle mogelijkheden van zo'n duo benut. Het tekstverwerkingspakket SIGNUM is helemaal geschreven voor de Atari ST en een Nec-Pinwriter. Het is op dit moment het enige pakket dat een Nec-24 naalddriver bevat. Op deze manier krijgt u een perfect geprinte tekst die u alleen nog maar met de duurdere Laser-printers kunt vergelijken.

Dat de combinatie Atari ST 1040, een Nec-Pinwriter met 24-naaldstechnologie en het tekstverwerkingspakket SIGNUM perfect is, bewijst alleen al het feit dat in de twee eerste weken van dit jaar 600 pakketten SIGNUM verkocht zijn aan de professionele gebruiker.

Wanneer u zich echt wilt laten overtuigen van alle mogelijkheden zoals het kunnen werken met maar liefst zeven fonts tegelijkertijd, bezoek dan uw geautoriseerde Atari ST dealer, die u met het pakket SIGNUM het desktop publishing tijdperk zal binnenleiden.

Het pakket SIGNUM wordt in Nederland uitgegeven door P.D.S. en kost f. 448,— ex. B.T.W.

Distributeur:

**SOFTPACK**  
INTERNATIONAL  
Postbus 6250,  
2702 AG Zoetermeer.  
Tel.: 079-412563



# CROSS-REFERENCE

Een gevolg van de populariteit van GfA-BASIC is dat veel mensen grotere programma's gaan schrijven. Grote programma's gebruiken veel variabele- en procedurenamen. Om dubbel gebruik te voorkomen, is het makkelijk als daar een lijst van gemaakt kan worden.

Bij de GfA-interpreter wordt een programma geleverd dat dit zou moeten kunnen. Helaas werkt het vrij omslachtig. De gebruiker moet het programma als tekstbestand wegschrijven voor het geanalyseerd kan worden. Peter Witvoet ging op zoek naar een snellere manier.

## RÖNTGENFOTO

Om een programma te kunnen schrijven dat een lijst van alle gebruikte variabelen produceert, moeten we allereerst weten hoe de GfA-editor programma's opslaat. Om dit te onderzoeken maakte ik gebruik van het programma TEST.BAS. Dit programma bevat alle soorten variabelen, procedures en labels, zodat we naar een systeem in de manier van opslaan kunnen gaan zoeken. Een run van TEST.BAS is weinig zinvol. Het programma is slechts kleurstof voor de röntgenfoto die we gaan maken.

Nadat ik het programma TEST.BAS op schijf gezet had, moest ik met een hulp-programma aan de slag. MichTron Utilities bleek hier uitermate geschikt voor. De twee afdrucken van MichTron tonen ons de interne opslag van TEST.BAS.

Enige basiskennis van het hexadecimaal stelsel is vereist om de structuur in het programma te kunnen volgen. TEST.BAS is grofweg in drie blokken te verdelen: een blok pointers met de variabele-en-procedurenamen, het programma en een blok gegevens voor de interpreter. Het eerste blok is de plaats waar we meer over willen weten.

## POINTERS

De allereerste byte geeft aan hoe het programma is weggeschreven. Is de byte 00H (H=hexadecimaal), dan gaat het om een normale SAVE. Programma's die met PSAVE zijn weggeschreven, hebben hier een andere waarde staan. Alle variabele- en procedurenamen zijn dan vervangen door pointers, zodat dergelijke programma's voor onze cross-reference afvallen. De tweede byte geeft aan om welke GfABASIC-versie het hier gaat. Op dit ogenblik zal dit byte de waarde 02H hebben (versie twee).

De volgende acht bytes bevatten de naam 'GfABASIC', zodat de interpreter kan zien dat het bestand een echt GfA-programma is.

De vier bytes hierna worden niet gebruikt.

Nu komen de pointers aan de beurt. Deze zijn telkens als vier-byte waarden opgeslagen. Normaal gesproken hebben de eerste twee bytes de waarde 00H. De derde en vierde byte verschaffen ons informatie over de plaats waar een variabelenreeks zich bevindt. Na bestudering blijkt dat er twaalf pointers gebruikt worden. De volgende lijst geeft hier duidelijkheid over:

- Pointer 1: real variabelen
- Pointer 2: string variabelen
- Pointer 3: integer variabelen
- Pointer 4: boolean variabelen
- Pointer 5: real arrays
- Pointer 6: string arrays
- Pointer 7: integer arrays
- Pointer 8: boolean arrays
- Pointer 9: labels

Pointer 10: procedures  
Pointer 11: deffn  
Pointer 12: (onbekend)

```

      ..... TEST
      A=5
      B=6
      C$="1"
      D$="2"
      E=1
      F=0
      G=True
      H=False
      Dim I(5),J(6),K(1,2),L(3,4)
      Dim M$(1),N$(1),O$(1,1),P$(1,1)
      Dim Q$(1),R$(1),S$(1,1),T$(1,1)
      Dim U!(1),V!(1),W!(1,1),X!(1,1)
      Goto Start
      A=A
      Start:
      A=A
      Gosub Sub
      Gosub Par(1,"t")
      Restore Gegevens
      Print Fn A
      End
      Procedure Sub
      Local Lok,Lokaal,Lok$,Lokaal$
      Return
      Procedure Par(Aa,Aa$)
      Dim Plaatselijk(3)
      Return
      Gegevens:
      Data TAURUS SOFTWARE
      Deffn Drie=3*A
      .....

```

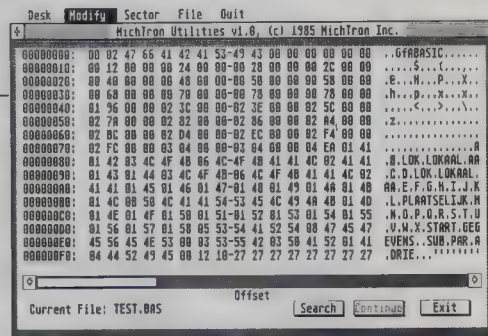
Hierna volgen nog zestien vier-byte pointers. Waar deze voor gebruikt worden, is me op dit moment nog niet duidelijk. Als u zelf een dump maakt, is het mogelijk dat u hier andere waarden aantreft dan ik.

Direkt achter deze pointers begint de lijst van namen waar de eerste twaalf pointers naar wijzen. De eerste pointer heeft de waarde 0012H. Dit is 18D (decimaal). Dit betekent dat de eerstvolgende 18 bytes de namen van alle real variabelen bevatten. We komen uit ons testprogramma A, B, LOK, LOKAAL en AA tegen. Nu blijkt dat de namen van algemene en lokale variabelen gewoon bij elkaar gezet worden. We kunnen ze dus niet uitsplitsen. Als we goed kijken, zien we dat er voor elke variabelenaam een byte staat. De byte

geeft aan uit hoeveel bytes de naam bestaat.

De tweede pointer geeft de byte staat van de stringvariabelen. Het is de bouw als bij de real variabelen. De tweede pointer heeft als waarde (2\*16+4=36D). Dit betekent dat de stringvariabelen is dus 36-18=18H. De integerlijst is 28H.

Enzovoort... Heeft een pointer de waarde 0000, dan is de pointer hoort niet in het voor.



## POINTERS NAAR POINTERS

Wie de afdrucken van MichTron verder bekijkt, ziet dat in het programma verder geen namen van variabelen en procedures voorkomen. Dat hoeft ook niet, want de gebruikte namen worden vervangen door pointers. En ook hier is systeem in te ontdekken. We weten dat we minstens elf pointers voorin hebben staan. De GfA-editor heeft uitgezocht om welk type het gaat en maakte een verwijzing. De verwijzing bestaat uit het cijfer E (hex) en een volgnummer. 0 voor real variabelen, 1 voor stringvariabelen, enzovoort. Komen we E0 tegen, dan verwijst deze pointer naar de lijst van realnamen. Dan moet de interpreter nog weten welke naam bedoeld wordt. Ook hier een volgnummer in hex. Kom je E0 02 tegen, dan zoek je bij de real variabelen naar de derde naam, hier: LOK. Denk erom: deze pointers beginnen bij 0 te tellen!

## SPOOKVARIABLEN

Het kan gebeuren dat er variabelenamen, enz. in de lijst staan die niet in het programma voorkomen. Deze namen zijn echter wel eens eerder gebruikt, maar naderhand door de programmeur veranderd of gewist. De editor wist of verandert geen namen in de lijst, maar laat ze gewoon staan. De enige manier om deze namen te verwijderen is de volgende: save het programma met save.a. Wis het programma en haal het met merge weer naar binnen. Uw programma wordt meteen korter.



theo  
de i  
progr.  
fabetisch  
procedures  
progr.  
progr.  
progr.

CROSS.BAS brengt de  
CROSS.BAS haalt  
uit een willekeurig  
op en print een al-  
an variabelen, labels,  
s. Zorg ervoor dat de  
n staat voor u  
ng stelt.  
riabelen is als volgt

: geen toevoegsel  
: \$  
: %  
: !  
: ()  
string array : \$()  
integer array : %()  
boolean array : !()

Het programma CROSS.BAS bestaat uit een hoofdprogramma dat in de juiste volgorde een aantal procedures aanroept:

#### Procedure Initieer:

De te gebruiken arrays worden in dit deel van het programma gedimensioneerd. P() voor de waarden van de pointers, T\$() voor

de te printen toevoegingen aan de namen, S\$() voor de pointertypes bij de namen in N\$() en tenslotte Sort\$() en Sort\$() voor het sorteren van de namen.

#### Procedure Kies.file:

Eerst moet het programma weten in welke drive de schijf met het te onderzoeken programma zit. Daarna wordt het programma door middel van een fileselectbox gekozen.

#### Procedure Check.basic:

Hier wordt gekeken of het programma een GfA-programma is en of het niet met Psave is weggeschreven.

#### Procedure Lees.pointers:

De waarden van de twaalf pointers worden ingelezen, naar decimaal omgerekend en opgeslagen in P(). Tevens worden eenendertig onbekende pointers ingelezen, zodat het programma uitkomt op de plaats waar de namenlijst begint.

#### Procedure Lees.namen:

Deze leest de namen waar de pointers

naar verwijzen in N\$(). Het type wordt vermeld in S\$().

#### Procedure Sorteert.namen:

Dit gedeelte van het programma zet de types en de namen in Sort\$(). Het type wordt in de eerste twee plaatsen vermeld, gevolgd door een spatie. Daarna volgt de procedure sorteert.

#### Procedure Print.namen:

Deze routine print alle namen in alfabetische volgorde onder de goede rubrieksnaam. Tussen de namen staan twee spaties en namen die niet op een regel passen, worden op een nieuwe regel geplaatst.

#### Procedure Lees:

Lees haalt een teken van de schijf. I bevat de ASCII-waarde van het laatst ingelezen teken. X\$ bevat de tekst als er meer tekens ingelezen zijn. L is een lokale variabele die aangeeft hoeveel tekens er van de schijf gelezen moeten worden.

```

*** VAR- EN PROC LISTER ***
Gosub Initieer
Repeat
  Repeat
    Gosub Kies.file
    If Naam$=""
      End
    Endif
    Gosub Check.basic
    If Kies=2
      End
    Endif
  Until Prima=True
  Gosub Lees.pointers
  Gosub Lees.namen
  Gosub Sorteert.namen
  Gosub Print.namen
  Gosub Kies(2,"Wilt U stoppen, of do =>
  orgaan ?")
  Close
  Until Kies=2
End
*** INITIEER ***
Procedure Initieer
  Dim P(12),T$(12),S$(1000),N$(1000), =>
  Sort$(1000),Sort$(1000)
  For X=1 To 12
    Read T$(X)
  Next X
  Data ,$,X,!,(),$( ),X( ),!{ },,,,
  Basic$="GfABASIC"
Return
*** KIES FILE ***
Procedure Kies.file
  Clr I$
  Print At(10,10);"Welke drive (A-?) ?"
  Form Input 1 As I$
  If Len(I$)=0
    I$="E"
  Endif
  Disk=Asc(Upper$(I$))-64
  Chdrive Disk
  D$=I$+"*.BAS"
  Fileselect D$,"",Naam$
  If Naam$="" And Naam$<>"\"
    Open "i",#1,Naam$
  Endif
Return
*** CHECK BASIC ***
Procedure Check.basic
  Kies=1
  Gosub Lees(1)
  Beschermd=I
  Gosub Lees(1)
  Versie=I
  Gosub Lees(8)
  If X$=Basic$
    Prima=True
    If Beschermd<>0
      Prima=False
    Gosub Kies(1,"Dit programma is =>
  met PSAVE!weggeschreven. Hier kan ik! =>
  geen lijst van maken!")
  Close
Endif

```

```

Else
  Prima=False
  Gosub Kies(1,"Dit is geen GfABASI =>
  C-programma!")
  Close
Endif
Return
*** KIES ***
Procedure Kies(I,I$)
  Alert I,I$,1,"DOORGAAN:STOPPEN",Kies
Return
*** LEES ***
Procedure Lees(L)
  Clr X$
  For X=1 To L
    I$=Input$(1,#1)
    X$=X$+I$
    I=Asc(I$)
  Next X
Return
*** LEES POINTERS ***
Procedure Lees.pointers
  Gosub Lees(6)
  Arrayfill P(),0
  For Lus=1 To 12
    Gosub Lees(1)
    Hb=I*256
    Gosub Lees(1)
    P(Lus)=Hb+I
    Gosub Lees(2)
  Next Lus
  For Lus=1 To 31
    Gosub Lees(2)
  Next Lus
Return
*** LEES NAMEN ***
Procedure Lees.namen
  Lijst=0
  For Lus=1 To 12
    Veldlengte=P(Lus)-P(Lus-1)
    Teller=0
    If Veldlengte>0
      While Teller<Veldlengte
        Inc Teller
        Gosub Lees(1)
        If I>0
          Teller=Teller+1
          Gosub Lees(1)
        Inc Lijst
        S$(Lijst)=Lus
        N$(Lijst)=X$
      Else
        Inc Teller
      Endif
    Wend
  Endif
  Next Lus
Return
*** SORTEER NAMEN ***
Procedure Sorteert.namen
  Sort$(0)=Str$(Lijst)
  For X=1 To Lijst
    If S$(X)<9
      Sort$(X)="00 "+N$(X)+T$(S$(X))
    Else
      Sort$(X)=Right$( "00"+Str$(S$(X)) =>

```

```

),2)+" "+N$(X)
  Endif
  Sort$(X)=X
  Next X
  Gosub Sorteert
  For X=1 To Lijst
    S$(X)=Val(Left$(Sort$(Sort$(X)),2))
    N$(X)=Right$(Sort$(Sort$(X)),Len( =>
  Sort$(Sort$(X))-3)
  Next X
Return
*** SORTEER **
Procedure Sorteert
  Local XX,MZ,HX,VX,Lengte%
  Lengte%=Lijst
  MZ=Int(Lengte%/2)
  Repeat
    For XX=1 To Lengte%-MZ
      HX=XX
      Repeat
        VX=HX+MZ
        Exit If Sort$(Sort$(HX))<Sort =>
$(Sort$(VX))
        Swap Sort$(HX),Sort$(VX)
        HX=HX-MZ
      Until HX<=0
    Next XX
    MZ=Int(MZ/2)
    Until MZ=0
  Return
*** PRINT NAMEN ***
Procedure Print.namen
  Local L
  Lprint "PROGRAMMA: ";Naam$
  Lprint
  Lprint "Variabelen:"
  For X=1 To Lijst
    If S$(X-1)<9 And S$(X)=9
      Lprint
      Lprint
      Lprint "Labels:";
      L=80
    Endif
    If S$(X-1)<10 And S$(X)=10
      Lprint
      Lprint
      Lprint "Procedures:";
      L=80
    Endif
    L=L+Len(N$(X))+2
    If L>79
      L=Len(N$(X))+2
      If L<79
        Lprint
      Endif
      Lprint N$(X);" ";
    Next X
    Lprint
  Return

```



# 6 REDENEN OM U OP START TE ABONNEREN

hebben we u nu al gegeven.

En we blijven doorgaan. Elk nummer opnieuw zullen we u bestoken met nieuwtjes, interviews, recensies, documentatie, programma's en voordelige aanbiedingen.

Als Amor zoveel pijlen afvuurt, moet hij toch eens een keer de roos treffen?

Een jaarabonnement bestaat uit zes opeenvolgende nummers en kan elk nummer ingaan.

Maak f 39,95 over naar postgiro 1962230

t.a.v. Divo computerservice/START  
M. Gijzenburg 14  
2907 HG Capelle a/d IJssel

Voor België:

Maak Bfr 800 over naar postgiro 000-1592758-18  
t.a.v. Divo computerservice/START  
M. Gijzenburg 14  
2907 HG Capelle a/d IJssel  
Nederland

Geef duidelijk aan welk nummer u als eerste wenst te ontvangen.

*P.S. Dat waren we u nog schuldig: Meneer X. (zie START 5) is overstag!*



## START LEZERSSERVICE

De listings in elk nummer van START zijn ook op diskette verkrijgbaar. U bespaart uzelf veel tijd voor betrekkelijk weinig geld.

Listings in een compiler-afhankelijke programmeertaal worden ook in hun uiteindelijke vorm op de START-diskette gezet.

Als u zelf niet over een compiler beschikt en het programma toch graag in uw bezit wilt hebben, kan dit een uitkomst zijn.

## NIEUW:

### START SPECIAL DISK 4: SPELLBOUND

(auteur: Rob Koopman)

Spellingschecker met uitgebreid Nederlands woordenboek. Controle tijdens het typen of achteraf. Toevoegen aan woordenboek mogelijk. Voor hoge of middelste resolutie.

### START SPECIAL DISK 5: SAPRISTI

(auteur: A. Bollongino)

Etikettenmanipulator met mogelijkheden voor adreslabels, diskettelabels en eigen ontwerpen. Inclusief database en RAM-disk. Voor hoge of middelste resolutie.

ETIKETTEN  
met een vrije layout  
u maakt ze eenvoudig en snel  
met  
SAPRISTI

### START DISK 6:

GfA BASIC: 1ST-aid, accload, font-editor, fontmanipulatie, floptest, astro, cross, bommelding, invoer. C: fontmanipulatie. 68000: fontmanipulatie, bommelding, cursusvoorbeeld. Pseudo: spellingschecker.

Verder zijn nog verkrijgbaar:

### START SPECIAL DISK 1: ELDICAD

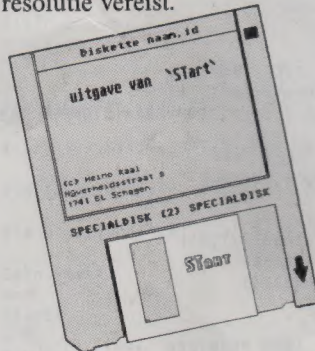
(auteur: Rob Hoo)

Programma waarmee onder muisbesturing elektronikaschema's gemaakt kunnen worden. Hoge resolutie vereist.

### START SPECIAL DISK 2: KATALOGER

(auteur: Heino Kaal)

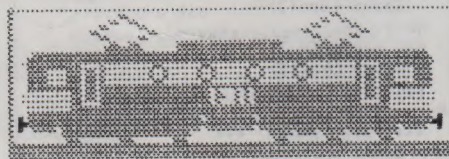
Programma waarmee u al uw directories kunt verzamelen, sorteren en afdrukken. Hoge resolutie vereist.



### START SPECIAL DISK 3: PAINT

(auteur: Anton Raves)

Icoon gestuurd tekenprogramma voor alle resoluties. Compatibel met DEGAS, Art Director, Doodle en Neochrome.



### START DISK 1:

BASIC: Go, ST-Base, muis, venstertitel en alertbox. Logo: spiraal, boom, schoepenrad en roos. C: 36-en. ASCII: Epson printer driver.

### START DISK 2:

BASIC: ST-tekst, klokreset, mastermind. LOGO: ijspret. 68000: klokreset. C: quicksort, maxi-format.

### START DISK 3:

GfA BASIC: ST-change, rename, menudemo, sorteerdemo, kalenders, run-only. ST BASIC: noodrem. LOGO: procedurebibliotheek. 68000: cursusvoorbeeld, noodrem.

### START DISK 4:

GfA BASIC: dirsor, kunST, sprites, recursie, run-only. ST BASIC: schaakeditor. LOGO: Dokter Logo. C: dubbelscherm en GEM-demo. 68000: cursusvoorbeeld.

### START DISK 5:

GfA BASIC: 3-D frames, BTW, gokkast, sounds, klok, ramdisk, run-only. LOGO: infopaneel. C: file-selector, toonladder. 68000: klok, ramdisk.

### A. SCREENDUMP

(auteur: Rob Koopman)

Uitgewerkt voorbeeld in Modula-2 om een eigen screendump routine te schrijven.

### B. MEGAMAX

(auteur: Egon Lubbers)

Programma's en sources van: Impossible Figures, Mastermind, Life en Music Construction Set.

### C. MAKE

Utility waarmee programmeren in C gemakkelijker wordt gemaakt. "Make" stuurt alleen gewijzigde programmadelen naar de compiler.

### D. ARC

Utility waarmee u uw programma's in sterk verkorte vorm kunt bewaren. Ideaal voor een zo klein mogelijk aantal backup-schijven.

### HARDWARE: TELTRON EXTENDER

Schakelkastje waarmee het populaire Teltron C64 modem op de ST aangesloten kan worden. Inclusief voorbeeldprogramma softwarematige baudrate instelling en autodial.

### PRIJZEN:

DISKS 1 t/m 6, A t/m D:

Abonnees	f 15,-
Niet-abonnees	f 25,-
KATALOGER/PAINT/SAPRISTI:	
Abonnees	f 39,-
Niet-abonnees	f 49,-
ELDICAD/SPELLBOUND:	
Abonnees	f 49,-
Niet-abonnees	f 69,-
TELTRON EXTENDER	f 199,-
START (oude nummers)	f 6,95

### BESTELWIJZE:

Maak het verschuldigde bedrag over op postgiro 1962230  
t.a.v. Divo Computerservice/START  
M. Gijzenburg 14  
2907 HG Capelle a/d IJssel  
onder vermelding van de gewenste producten.



# Atari ST..... ideale computers

Atari is de vriendelijke computer voor een vriendelijke prijs: een ideale combinatie. Nu levert Atari een complete range computers met een geheugen tot 4 Mb RAM in de Mega ST. Computers die overal inzetbaar zijn, bedrijfsmatig, particulier en voor verenigingen. Met Atari kunt u alle kanten op! Compudress maakt deze computers compleet met een Epson, Brother of Nec printer én de benodigde software. De toch al lage aanschafprijs behoeft geen probleem te zijn: U maakt gebruik van de financiële leasing met eigendoms-overdracht, die Compudress u biedt tegen gunstige tarieven.

## Prijsverlaging ! Atari 1040 STf bundel

Atari 1040 STf computer.  
1 Mb Ram geheugen.  
Ingebouwde diskdrive 720 Kb.  
muis (=gebruikersvriendelijkheid)  
monochrome monitor SM125.  
Basic programmeertaal  
Epson LX-800 dot matrix printer  
**Aktieprijs f 2798,-**  
**Financiële leasing v.a. f 66,53**

## Atari 520ST+ bundel

Atari 520 ST+ computer  
1Mb Ram geheugen  
diskdrive, muis  
monochrome monitor SM125,  
Tos/Gem in ROM  
ST-Basic programmeertaal  
2 public domain disks  
Inkl. 'Het AtariST boek'.  
**Speciale prijs f 1.498,-**  
**Financiële leasing v.a. f 37,94**  
*Zolang de voorraad strekt!*

## Epson en Nec printers

Epson LX-800 *Nieuw!*  
dot-matrix printer.  
**180 kar./sec.** In draft, N.L.Q.  
Inklusief tractor feeder en single  
sheet feeder guide.  
**Speciale prijs .....Bel**

Nec P6 dot-matrix printer  
24 naalds, 216 k.p.s. In draft  
72 k.p.s. In **Letter Quality!**  
8 Kb Input-buffer  
**Financiële leasing v.a. f 42,34**

## Atari Mega ST bundel !

Atari Mega ST computer.  
**2 of 4 Mb Ram.**  
Motorola 68000 processor  
ingebouwde diskdrive 720 Kb  
muis (=gebruikersvriendelijkheid)  
monochrome/kleuren monitor  
apart toetsenbord  
real-time klok  
TOS/GEM in ROM  
Nederlandstalige handleiding  
Epson LX-800 dot-matrix printer  
**Nú uit voorraad leverbaar!**  
**Prijs.....Bel 03481-2124**

## Nieuw! Atari PC

Atari PC computer  
512 Kb Ram  
8088-2 microprocessor  
8 en 4.77 MHz (omschakelbaar)  
ingebouwde 5.25 inch diskdrive  
hoge resolutie EGA monitor  
grafische modes: EGA, CGA  
Hercules, stand. monochroom  
serieel en parallel interface  
geïntegreerde muis-interface  
Nederlandstalige handleiding  
**Zie de Intekenaktie!**

*Bel voor de Atari-Infomappen!*

**\*Financiële leasing  
(12 tot 60 maanden)  
met eigendomsover-  
dracht voor f 10,-.  
Oók voor particulieren!**

**Kopers zijn uiteraard ook welkom!**

## Officiële dealer van o.m.:

Apple, Atari, Epson, Olivetti,  
Commodore Amiga, Tulip,  
Schneider PC.  
Printers: Brother, Epson en Nec.  
Software: AS Plus, Grote Beer, Kern  
programma's, Softkey e.v.a..

## Bezoek showrooms (700 m²) en demonstratie na afspraak.

Kamerik ligt centraal in Nederland  
tussen Woerden en Breukelen. (3 km  
van Woerden, 15 km van Oudenaarden.)  
Levering doorgaans uit voorraad.  
Garantie en service binnen 24 uur.  
Eigen technische dienst.  
Eigen trainings/opleidingsfaciliteiten.  
**Prijzen zijn inclusief 20% BTW.**  
(Deze advertentie is gemaakt op  
een Apple Macintosh DTP systeem)

**Kamerik**  
(bij Woerden)



## Atari Intekenaktie !

**Wilt u als één van de eersten  
in Nederland in het bezit  
komen van een Atari PC?**

**U kunt nu intekenen!  
Vraag het aktieformulier.  
Bel 03481-2124**

**Onze financiële leasing op  
gunstige voorwaarden blijft  
ook geldig tijdens de aktie.**



**Nieuw:  
Art Director en Film Director.**

PDS brengt weer nieuwe pakketten om de Atari ST nieuwe mogelijkheden te geven. Met Art Director en Film Director krijgt uw creativiteit vleugels.

# ONTDEK UW CREATIEVE TALENT MET DE UNIEKE PDS PAKKETTEN.

Art Director biedt u de mogelijkheid om met maar liefst 76 functies de fraaiste kleurenillustraties te maken. Het programma voorziet in gehele en/of gedeeltelijke vervormingen of kleurwijzigingen van de illustraties. Veel verschillende technieken zoals schilderen, airbrush en tekenen kunt u toepassen. Nooit eerder was er zo'n compleet tekenprogramma!

Film Director zet u op de stoel van de filmregisseur. Elk filmgenre dat u wenst kunt u maken, en is vergelijkbaar met een professionele tekenfilmproductie. U produceert een perfect vloeiend overlopende tekenfilm waarin u achtergronden naar keuze creëert. Met het programma kunt u monteren, snellopen, teruglopen, ja zelfs muziek toevoegen. Of de aftiteling van uw videofilm verzorgen.

Voorals u Film Director combineert met Art Director zijn de mogelijkheden onbegrensd. En met behulp van de apart te leveren video-interface kunt u de film overzetten op videoband. Film Director en Art Director, twee unieke PDS pakketten die uw dealer u graag laat zien.

**PDS. De software voor de Atari Professional.**



## PDS

PDS, Postbus 5829, 2280 HV Rijswijk.